



Departamento de
Engenharia Electrotécnica

Desenvolvimento de um Sistema de Nivelamento Hidrostático Ultra-sónico Móvel

Relatório de Estágio apresentado para a obtenção do grau de
Mestre em Engenharia Electrotécnica

Autor

Cristóvão André Dionísio Barreto

Orientadores

Professor Doutor Adelino Jorge Coelho Pereira

Professor Doutor Paulo Jorge Ribeiro da Fonte

Instituto Superior de Engenharia de Coimbra

Supervisores

Eng.º Cédric Charrondière

Eng.º Patrick Bestmann

Conseil Européen pour la Recherche Nucléaire

Coimbra, 20 de Outubro de 2015

AGRADECIMENTOS

A gratidão, tanto mais inarticulada é quanto profunda.

Sendo mais do que a soma de diversas partes, resta-me agradecer singelamente todo o apoio e direcção de todos ao meu redor, que acredito estarem presentes e reflectidos nestas palavras.

A minha gratidão, verdadeiramente memória do coração.

RESUMO

O presente Relatório de Estágio expõe o desenvolvimento de um sistema de Nivelamento Ultra-sónico Hidrostático - HLS de alta precisão, a ser realizado no *Conseil Européen pour la Recherche Nucléaire* – CERN. O projecto HLS consiste no desenvolvimento de um protótipo estado-da-arte para a medição da altura entre os diferentes pontos dos ímanes supercondutores constituintes do *Large Hadron Collider* – LHC durante o *Long Shutdown 1* - LS1, um parâmetro operacional crítico para factores de escalas totalmente díspares, desde o alinhamento do Feixe de Protões nos Colimadores como para a análise estrutural do túnel de 27 km. Os níveis de precisão mínimos almejados situam-se nos 50 μ m e devem ser atingidos preservando a mobilidade do sistema ao longo do LHC, possibilitando uma campanha de medições.

Além da descrição do processo e desenvolvimento de todo o projecto e sua metodologia empregue, este relatório conclui com a apresentação e análise dos resultados operacionais da aplicação do sistema realizados sobre o LHC, incluindo precisões atingidas, tempos de operação e outras considerações.

Palavras-chave: Nivelamento Hidrostático, Ultra-sónico, Móvel, Large Hadron Collider, Instrumentação

ABSTRACT

The present Internship Report exposes the development of a high precision prototype for a Mobile Ultrasonic Hydrostatic Levelling System - HLS, conducted in the *Conseil Européen pour la Recherche Nucléaire* - CERN. The HLS project consists in the development and operational testing of a state of the art innovative instrumentation system to allow the measuring of the height differences between the superconducting magnets of the Large Hadron Collider - LHC during the Long Shutdown 1 - LS1, a parameter deemed critical in various aspects, from structural analysis of the 27km tunnel to the alignment of the Proton Beam in the Accelerators. The target precision level is set at a minimum of 50 μ m and should be maintained while respecting the mobility of the system throughout the LHC, in order to establish a successful measurement campaign.

Apart from the description of the development process and all of its employed methodology, this report concludes with the presentation and analysis of the operational results the application of the system performed in the LHC, including the level of precision obtained, operation times and other remarks.

Key-words: Hydrostatic Levelling, Ultrasonic, Mobile, Large Hadron Collider, Instrumentation

ÍNDICE

Agradecimentos	i
Resumo	iii
Abstract.....	v
Índice.....	vii
Índice de Figuras.....	x
Índice de Tabelas.....	xiii
Abreviaturas.....	xiv
Simbologia.....	xvi
CAPÍTULO 1 - Introdução	1
1.1 – Considerações gerais	1
1.2 – Contexto do Estágio.....	2
1.2 – Estrutura e Organização do Trabalho.....	2
CAPÍTULO 2 – Conseil Européen pour la Recherche Nucléaire - CERN.....	5
2.1 – Considerações Gerais.....	5
2.2 - Large Hadron Collider - LHC.....	8
2.3 – Sistema de Coordenadas do CERN e Metodologia	11
2.4 – Especificações do Sistema	14
2.4.1 – Sistema de Nivelamento Hidrostático Ultra-Sónico	18
2.4.2 – Mesa de Translação: Zaber Dovetail.....	23
2.4.3 – Encoder Linear: Régua Óptica Renishaw	24
CAPÍTULO 3 – Projecto HLS.....	27
3.1 – Desenvolvimento & Implementação	27
3.2 – Ambiente de Desenvolvimento.....	28
3.3 – Projecto LabVIEW	29
3.4 – HLS_Main e a Arquitectura da Aplicação	30

3.4.1 – Estratégia de Tratamento de Erros.....	33
3.4.2 – Main_Lite.....	34
3.4.3 – Processos de Inicialização	35
3.4.3.1 – INI File Manager.....	36
3.4.4 – Eventos dinâmicos com o Módulo: Theo Data Viewer.....	38
3.4.5 – Queues	39
3.5 – Módulos Top-Level e Auxiliares	40
3.5.1 – Exit Manager.....	41
3.5.2 – Monitor de Estados - ‘Status Monitor’	42
3.6 – <i>Logger: Registo de Dados de Operação</i>	43
3.6.1 – Visualizador de Logs.....	44
3.7 – <i>Mesa de Translação: Motor Zaber Dovetail</i>	45
3.7.1 – HLS Motor Main.....	46
3.7.1.1 - Put in Place: Position from File.....	46
3.7.1.2 - Put in Place: Fine.....	49
3.7.1.3 – Movimento de Calibração	49
3.8 – <i>Régua Escala Óptica</i>	51
3.9 – <i>Sensor de Nivelamento Hidrostático Ultra-sónico (UHLS)</i>	53
3.9.1 – Tratamento Estatístico	54
3.9.2 – Gestor de Configuração do UHLS (Configuration Manager)	57
3.10 – <i>Tratamento de Erros – Central Error Handler</i>	60
3.11 – <i>Theo Data Viewer</i>	62
3.12 – <i>Publicação via Web: Conexões Remotas</i>	65
3.12.1 – Publicação Web do Front Panel.....	66
3.12.2 – Aplicação para Tablet.....	67
3.13 – <i>Módulo de Testes</i>	68
CAPÍTULO 4 – Análise de Resultados	71

<i>4.1 – Resultados Laboratoriais</i>	<i>71</i>
<i>4.2 – Resultados Operacionais: LHC.....</i>	<i>75</i>
CAPÍTULO 5 – Conclusões	78
<i>5.1 – Conclusões.....</i>	<i>78</i>
<i>5.2 – Desenvolvimentos Futuros.....</i>	<i>79</i>
Referências Bibliográficas	81

ÍNDICE DE FIGURAS

Figura 2.1 - Representação geográfica do CERN	5
Figura 2.2 – Detector Atlas (à esquerda) e representação do interior do LHC (direita)	6
Figura 2.3 – Diagrama de interligação dos aceleradores CERN	7
Figura 2.4 - Representação geográfica do complexo de aceleradores	8
Figura 2.5 - Princípio de funcionamento de Detector por captura de partícula	9
Figura 2.6 - Detecção do Bosão de Higgs pela experiência ATLAS (esquerda) e CMS (direita)	10
Figura 2.7 – Representação Ortometria: diferença entre ‘Z’ e ‘H’	11
Figura 2.8 – Representação Geóide para o CCS	12
Figura 2.9 – Íman Supercondutor LHC (Dipolo)	13
Figura 2.10 – Exemplo do princípio de Vasos Comunicantes	13
Figura 2.11 – Princípio de operação do sistema HLS	14
Figura 2.12 – Componentes de <i>hardware</i> principais do HLS	15
Figura 2.13 Componentes <i>Hardware</i> HLS. A azul, o Módulo <i>UHLS</i> , a amarelo o <i>Encoder</i> Óptico e a vermelho o Motor a Passo e mesa de translação	16
Figura 2.14 – Representação do <i>Survey Train</i>	17
Figura 2.15 – Características de sistemas de nivelamento hidrostático por tecnologia (esquerda) e degradação de sensores capacitivos expostos a água	18
Figura 2.16 – Módulo sensor <i>Ultrasonic Hydrostatic Levelling System</i>	19
Figura 2.17 – Esquemático da electrónica de funcionamento da MPE	20
Figura 2.18 - Peça de Referência em Quartzo (esquerda), esquemático de reflexões (centro) e análise das reflexões da técnica <i>Impulso-Eco</i> (direita) [7] [10]	21
A identificação dos factores da expressão (2.1) segue a denominação expressa pela Figura 2.19. Desta forma, o nível da água <i>H_w</i> é dado por:	21
Figura 2.20 – Acima: Reflexões em dados brutos. Abaixo: pormenor da reflexão R2	22
Figura 2.21 – Mesa de translação Zaber Dovetail e suas principais características	23
Figura 2.22 – Régua Óptica <i>Renishaw Optical Scale</i> e Adaptador BiSS-USB	24

Figura 2.23 – Diagrama de Funcionamento da Régua Óptica e suas principais características	25
Figura 2.24 – Telegrama de mensagem Protocolo BiSS	25
Figura 3.1 – Cronograma de desenvolvimento do projecto HLS.....	27
Figura 3.2 – Diagrama de funcionamento da ferramenta RADE	28
Figura 3.3 – Representação de Funcionamento da função <i>Main</i>	30
Figura 3.4 – GUI da <i>Main</i> , versão completa	31
Figura 3.5 – Representação da função <i>Main</i> e dos seus canais de comunicação (<i>queues</i>)	32
Figura 3.6 – Exemplo dos mecanismos de encaminhamento para tratamento de erros embutidos na infra-estrutura da aplicação.....	33
Figura 3.7 – GUI da função <i>Main_Lite</i>	34
Figura 3.8 – Exemplo de diálogo durante a procura de ficheiros de inicialização	37
Figura 3.9 – Fluxograma de funcionamento: <i>INI Manager</i>	37
Figura 3.10 – Representação do funcionamento das funções <i>Main</i> perante eventos dinâmicos com o módulo <i>Theo Data Viewer</i>	38
Figura 3.11 – Repartição de Queues por sector (esquerda). Procedimento de alocação de recursos em respeito à referência da Queue (direita)	39
Figura 3.12 – GUI <i>Status Monitor</i>	42
Figura 3.13 – Diagrama de operação do <i>Logger</i>	43
Figura 3.14 – GUI do Visualizador de Logs	44
Figura 3.15 – Princípio de Operação e Fluxograma de Funcionamento da função: <i>Put in Place: Position from File</i>	47
Figura 3.16 – NI DAQ USB 6009 e Electroválvula	48
Figura 3.17 – Diagrama de Operação da função <i>Put in Place: Fine</i>	49
Figura 3.18 – Fluxograma de Funcionamento: <i>Movimento de Calibração</i>	50
Figura 3.19 – Representação de Operação do Módulo de controlo da Régua Óptica	51
Figura 3.20 – GUI do Módulo <i>UHLS</i> , apresentando as ‘ <i>Opções Avançadas</i> ’.....	53
Figura 3.21 – Sequência de Operação do módulo <i>UHLS</i> ao efectuar uma medição.....	54

Figura 3.22 – Fluxograma com os procedimentos de Resolução de Erros	55
Figura 3.23 - Fluxograma dos procedimentos de Tratamento Estatístico das medições.....	56
Figura 3.24 – GUI do Gestor de Configurações do Módulo UHLS	57
Figura 3.25 – Exemplo da leitura de dados através da funcionalidade <i>Read Raw</i>	58
Figura 3.26 - Gráfico de Resultados de teste estático prolongado, apresentando variação do nível de líquido e evidenciando relação com ciclos de climatização.....	59
Figura 3.27 – Fluxograma de procedimento para o tratamento de Erros através do <i>Central Error Handler</i>	61
Figura 3.28 – Suporte de Encaixe (esquerda) e sua localização típica num Íman dentro do LHC	62
Figura 3.29 – GUI do Módulo <i>Theo Data Viewer</i>	63
Figura 3.30 – Acima: Exemplo das possibilidades de movimentação perante o diferente posicionamento das cápsulas. Abaixo: Fluxograma de operação do <i>Theo Data Viewer</i>	64
Figura 3.31 – Sistema HLS integrado, versão final colocado num <i>rack</i>	65
Figura 3.32 – Configurações exemplo para o <i>Web Server publisher</i>	66
Figura 3.33 – GUI da aplicação <i>Data Dashboard (esquerda)</i> e representação dos métodos na janela de projecto LabVIEW (direita)	67
Figura 3.34 – GUI do Módulo <i>Tester</i> e representação dos seus canais de comunicação.....	68
Figura 4.1 – Sistema HLS durante os testes de operação fina no Laboratório de Metrologia	71
Figura 4.2 – Pormenor do sistema HLS durante testes de funcionamento em inclinação (esquerda) e estação meteorológica <i>Thommen</i> para registo das variáveis ambientais	72
Figura 4.3 – Comportamento HLS durante medições prolongadas.	72
Figura 4.4 – Gráficos de Resultados de precisão: Série única (acima) e sequência de 100 séries (abaixo).	73
Figura 4.5 – Medição da diferença de nível entre dois pontos de um Íman no LHC.....	75
Figura 4.6 – Transporte do Sistema HLS através do LHC (esquerda) e Medição da diferença de nível entre dois pontos de um íman Quadripolo (direita)	76
Figura 7.1 – Representação Futura do sistema HLS de Medições.....	79

ÍNDICE DE TABELAS

Tabela 2.1 – Relação entre níveis de energia de um próton por acelerador	6
Tabela 2.2 - Factos Operacionais Gerais do LHC	9
Tabela 3.3 – Parâmetros BiSS utilizados	52
Tabela 3.4 – Formato dos dados A guardar segundo as normas do CCS.....	62
Tabela 4.1 – Parâmetros de operação empiricamente determinados.....	74

ABREVIATURAS

HLS – Hydrostatic Levelling System

CERN – Conseil Européen pour la Recherche Nucléaire

LHC – Large Hadron Collider

LS1 – Long Shutdown 1

ICE – Industrial Controls and Engineering

MTA - Measurements, Tests and Analysis

MPE – Metering Point Electronics

LINAC2 – Linear Accelerator 2

PS – Proton Booster

SPS – Super Proton Synchrotron

CCS – CERN Coordinate System

HMI – Human Machine Interface

ALARA – As Low As Reasonably Achievable

DESY – Deutsches Elektronen-Synchrotron

ADC – Analog-Digital Converter

UHLS – Ultrasonic Hydrostatic Levelling System

BiSS – Bidirectional Serial Synchronous

SSI – Serial Synchronous Interface

RADE – Rapid Application Development Environment

GUI – Graphical User Interface

VI – Virtual Instrument

CEH – Central Error Handler

Hw – Nível de Água

D1 – Parâmetro de Calibração da peça de Referência

R1 – Primeira reflexão da técnica Impulso-Eco

OF – Tempo de Propagação da reflexão

FGV – Functional Global Variable

SIMBOLOGIA

TeV – Tera Electrão-Volt

A – Ampere

mA – miliampere

V – Volt

mV – millivolt

Ω – Ohm

kV – Quilovolt

km - Quilómetro

m – Metro

mm – milímetro

μm - micrómetros

s – Segundo

ms – milissegundo

ns – nanosegundo

mrاد - miliradianos

MJ - MegaJoule

°C – Graus Centígrados

CAPÍTULO 1 - INTRODUÇÃO

1.1 – Considerações gerais

Este relatório segue um modelo lógico estabelecido que procura abordar a descrição da execução do projecto aproximada ao máximo com a ordem do seu desenvolvimento. Em primeira instância, será apresentado o contexto e as mecânicas do estágio em si, seguidas pela referência ao desafio técnico em questão e a necessidade da solução que deste se eleva. Após esta contextualização será apresentado a descrição mais detalhada da implementação concreta do projecto, incluindo a escolha da tecnologia e metodologia justificativa, desde o desenvolvimento de módulos e subsistemas a testes de operação. Finalmente, serão apresentados e analisados os resultados reais obtidos para que adicionem a sua contribuição tanto à descrição técnica como para corroboração do sucesso do projecto.

A maior relevância deste relatório incide precisamente no aspecto técnico de construção e desenvolvimento, juntamente com a sua implementação prática e as considerações que daí advêm baseadas na sua utilização no seu ambiente final, o *Large Hadron Collider* - LHC. Em síntese, procura-se expor todos os passos de desenvolvimento detalhadamente mas também contrapor com o processo iterativo que advém da obtenção dos resultados operacionais e subsequentes alterações e melhoramentos.

Embora seja mantida uma camada de abstracção teórica ao abordar alguns dos conceitos expostos por este relatório, toda a aplicação e código serão disponibilizados em anexo, bem como todas as especificações do *hardware* utilizado, permitindo uma visão o mais completa possível de todo o projecto.

Não obstante de terem sido realizadas outras funções e tarefas, como participação em outros projectos de menor escala e até suporte técnico a outras secções, o principal objectivo do estágio consistiu no desenvolvimento de um protótipo para a medição da diferença de altura entre os pontos dos ímanes supercondutores do LHC, designado por *Hydrostatic Levelling System* – HLS devido ao seu princípio de funcionamento e utilizado transdutores ultra-sónicos.

Durante o ano de 2014 e previsto até finais de Março de 2015 encontra-se a decorrer o período do *Long Shutdown 1* – LS1, onde o LHC encontra-se em operações de manutenção e ampliação e durante as quais o acesso ao mesmo é permitido. Os resultados deste projecto são então criticamente limitados por uma janela temporal impossível de ser movimentada e essenciais na calibração dos detectores, aceleradores e colimadores. O sistema deverá possuir uma precisão de pelo menos 50 μm , e manter um nível de mobilidade por forma a permitir que possa um operador realizar as medições ao longo de todo o LHC, integrando os resultados na base de dados adequada sob o formato correcto.

1.2 – Contexto do Estágio

Inserido no âmbito do programa curricular do Mestrado em Automação e Comunicações em Sistemas de Energia, a disciplina de estágio prevê que um projecto seja realizado pelo aluno num ambiente apropriado. Com este intuito, foi com grande felicidade que a aprovação da candidatura ao programa de *Technical Student* foi recebida, conduzindo ao estabelecimento do celebrado acordo entre o ISEC e o CERN.

Iniciando-se em Janeiro de 2014 e terminando em Dezembro do mesmo ano, o projecto de desenvolvimento de um protótipo ‘Estado-da-Arte’ para a medição da diferença de altura entre os pontos dos ímanes do LHC, designado por HLS (*Hydrostatic Levelling System*) devido à sua técnica de operação, foi desenvolvido no seio da Secção de *Measurements, Tests and Analysis*, do Grupo de *Industrial Controls and Engineering*, pertencente ao Departamento de Engenharia do CERN (EN-ICE-MTA). Ligados a este projecto, tornando-o possível e com todos os meus maiores agradecimentos, encontram-se os Professores Doutores Adelino Pereira, Paulo Fonte e Fernando Lopes, nas posições de Orientadores e Coordenador do Mestrado – Docente da Disciplina. Como Supervisores directos no CERN, Cédric Charrondière (EN-ICE-MTA), líder da secção MTA Adriaan Rijllard, e Patrick Bestmann, pertencente ao grupo de *Machines & Experimental Facilities*, secção de *Large Scale Metrology*.

1.2 – Estrutura e Organização do Trabalho

O presente relatório encontra-se dividido em 5 Capítulos centrais seguindo uma distribuição coerente com a estrutura de desenvolvimento.

No primeiro capítulo, é apresentada uma sintética introdução referindo as directivas gerais de todo o projecto, juntamente com o seu contexto e as mecânicas do estágio.

Seguidamente, no Capítulo 2 é aprofundado todo o cenário circundante ao mesmo, nomeadamente uma apresentação clara das tarefas e objectivos para o trabalho a desenvolver, conjugados a partir da problemática existente, a sua importância e necessidade de resolução. A partir destas directivas são expostas também os métodos e meios que serão utilizados, devidamente justificados fundamentalmente pela natureza da problemática em questão.

Após encontrar-se devidamente enquadrado e com objectivos e meios claramente definidos, o Capítulo 3 aborda de uma maneira mais técnica e específica o próprio desenvolvimento e implementação de todo o projecto, expondo todo o núcleo do trabalho desenvolvido, contraposto com as escolhas técnicas implementadas. Note-se que este capítulo apresenta os seus conteúdos partindo de um ponto de vista o mais elevado possível, omitindo algumas mecânicas básicas de funcionamento ou operação, visando ser o mais conciso e preciso possível, oferecendo uma visão mais clara de todo trabalho desenvolvido.

O Capítulo 4 apresenta os resultados práticos obtidos após o desenvolvimento prático e o processo iterativo de correcção e alterações originado das suas inferências. Esta exposição crítica dos resultados obtidos, seu significado e importância servirão de base para que no Capítulo 5 seja possível formar as conclusões finais sobre todo o projecto, actuando como os grandes indicadores do verdadeiro nível de sucesso do mesmo. O Capítulo 5 apresenta ainda as considerações futuras que serão implementadas e a relevância do trabalho desenvolvido em diversos projectos futuros.

CAPÍTULO 2 – CONSEIL EUROPEEN POUR LA RECHERCHE NUCLEAIRE - CERN

2.1 – Considerações Gerais

Situado na fronteira Franco-Suíça junto a Genebra, conforme visto na Figura 2.1, o CERN foi estabelecido em 1954 e festeja actualmente os seus 60 anos. A sua origem, conforme descrita por Rolf Heuer, Director-Geral do CERN é uma de como “um pequeno grupo de visionários que identificaram na Pesquisa Fundamental da ciência um veículo de união que permitiria reconstruir um continente e fomentar a união e a paz entre as nações” [1].



Figura 2.1 - Representação geográfica do CERN

A ideia da Ciência como uma linguagem universal não encontra melhor representação, com perto de 13.000 pessoas de 80 nacionalidades a construir com uma tapeçaria de tecido humano que representa um dos maiores e mais bem-sucedidos exemplos de cooperação mundial estabelecidos.

É fundamentada neste espírito que a sua missão pode ser descrita na sua melhor definição:

- **Tecnologia:** Expandir as fronteiras do conhecimento
- **Colaboração:** Aproximação das nações através da ciência
- **Educação:** Treinando os cientistas de amanhã

Utilizando os maiores e mais complexos instrumentos científicos do mundo na busca e desenvolvimento da *Pesquisa Fundamental*, olhando tanto para a origem do Universo como para os constituintes da matéria, a própria natureza da pesquisa é de tal forma magnânima e universal que ressoa com o espírito de cooperação internacional.

É dentro destes colossais instrumentos, como os apresentados na Figura 2.2, que encontramos os **Aceleradores** – que transportam e colidem as partículas a velocidades próximas da velocidade da luz, e os **Detectores**, que monitorizam e observam o acontecimento.

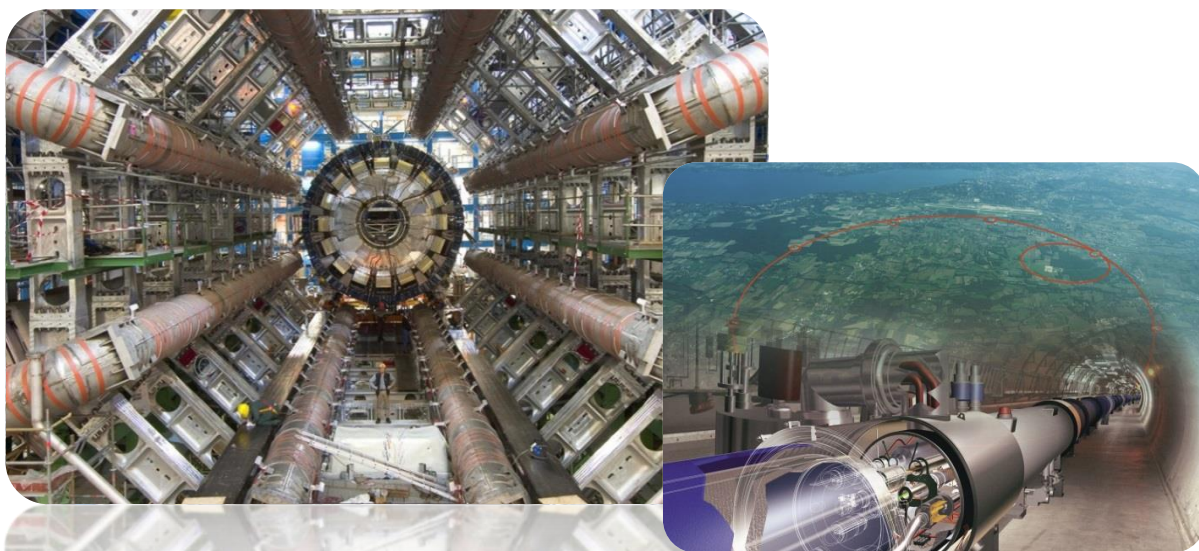


Figura 2.2 – Detector Atlas (à esquerda) e representação do interior do LHC (direita)

Embora nenhuma partícula possa viajar mais depressa do que a velocidade da luz no vácuo, não existe limite sobre quantidade de energia que essa partícula pode atingir. De facto, na física de partículas é este aspecto que ganha maior relevância: como exemplo, conforme a relação entre a velocidade relativa e energia de um protão exposta na Tabela 2.1, a velocidade de entrada de uma partícula no LHC é de 99,9997828% da velocidade da luz, com uma energia de 450 GeV, mas na sua velocidade máxima o aumento é mínimo, para 99,9999991%, mas com uma energia muito superior, na ordem dos 7 TeV [2] [3].

Tabela 2.1 – Relação entre níveis de energia de um protão por acelerador

Kinetic energy of a proton (K)	Speed (%c)	Accelerator
50 MeV	31.4	Linac 2
1.4 GeV	91.6	PS Booster
25 GeV	99.93	PS
450 GeV	99.9998	SPS
7 TeV	99.9999991	LHC
Relationship between kinetic energy and speed of a proton in the CERN machines. The rest mass of the proton is $0.938 \text{ GeV}/c^2$		

Os conceitos de Energia e Massa são, segundo a famosa equação de Einstein, $E = mc^2$, **comutáveis**, podendo um converter-se no outro [3][4]. Esta transformação ocorre no LHC a cada colisão com esta equivalência a permitir assim a medição de ambas as grandezas com uma só unidade, o já referido *Electrão-Volt*. De facto, as densidades energéticas atingidas pelas colisões no LHC são comparáveis às que ocorreram momentos após o *Big Bang*, através das quais construímos uma melhor percepção da evolução do Universo.

O Complexo de Aceleradores presente no CERN é uma sucessão de máquinas com *outputs* de energias de feixe cada vez maiores, injectando o mesmo na seguinte que o eleva um pouco mais além, e assim sucessivamente.

A sequência de operação, com a interligação entre os Aceleradores representada na Figura 2.3 pode ser descrita sucintamente da seguinte forma: são obtidos os prótons a partir de átomos de hidrogénio, que iniciam o seu percurso no primeiro acelerador, o *Linear Accelerator 2 - LINAC2*. Este por sua vez injecta-os no *Proton Booster - PS*, que os acelera até uma energia de 1,4 GeV. O agora denominado *Feixe* é introduzido então no *Super Proton Synchrotron - SPS*, e acelerado até 450 GeV, seguindo finalmente para o último acelerador na sequência, o *Large Hadron Collider - LHC* onde atingem um nível de energia final na ordem dos 7 TeV por feixe, que resulta num máximo de 14 TeV por colisão. Este valor em si não é representativo por si só, já que, embora a energia total do feixe seja de 350 MJ (equivalente a um comboio de 400 t), um simples bater de palmas representa uma colisão de maior *nível energético* do que a colisão dos feixes – 1 TeV representa o voo de um mosquito; mas é de facto a *Concentração Energética* destas colisões em espaços milhões de milhões de vezes mais pequenos que as tornam alvos de estudo [2].

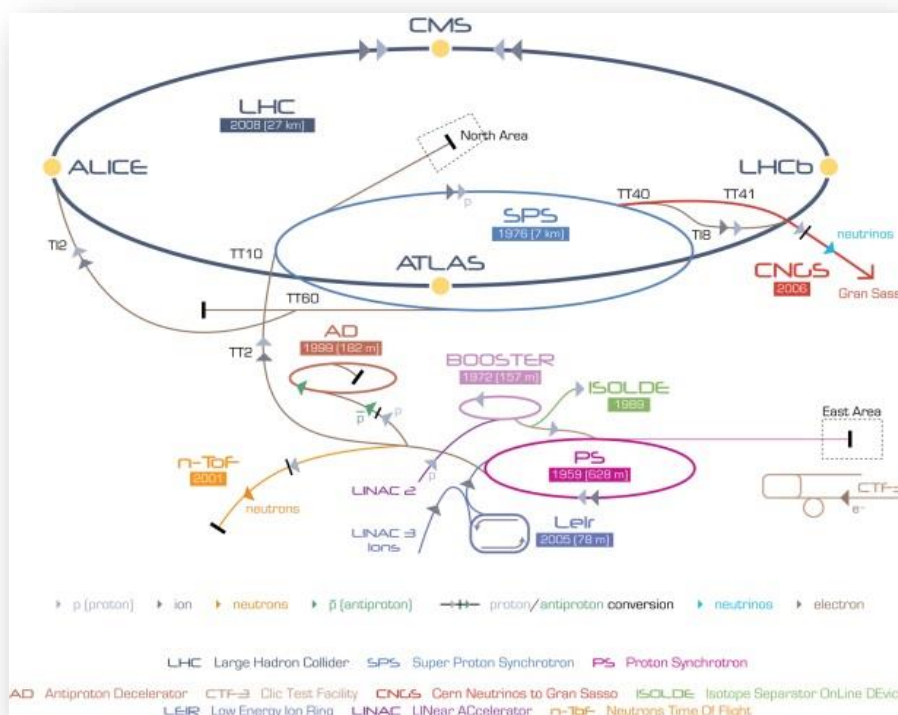


Figura 2.3 – Diagrama de interligação dos aceleradores CERN

2.2 - Large Hadron Collider - LHC

Com aproximadamente 27 km de extensão, o LHC mantém na verdade *dois* feixes de grupos de partículas, viajando em direcções opostas, e estabelece as colisões entre ambos em quatro pontos distintos, a uma profundidade que varia entre os 50 m e os 175 m abaixo da superfície, conforme a representação da Figura 2.4, e possuindo uma inclinação de 1,4 %. Sendo obrigatório manter o comprimento total do LHC por forma a não comprometer os níveis de energia atingidos pelas colisões, esta inclinação foi forçada pela falta de espaço existente no plano horizontal por forma a evitar formações rochosas demasiado duras. De facto, esta situação originou incríveis desafios de engenharia relativos à construção e operação do LHC. Como exemplo, a sua influência no desenvolvimento do sistema de bombas de circulação do hidrogénio, imperativo para manter os supercondutores no seu estado operacional, e dado a sua natureza, bastante vulnerável a estes aspectos construtivos.

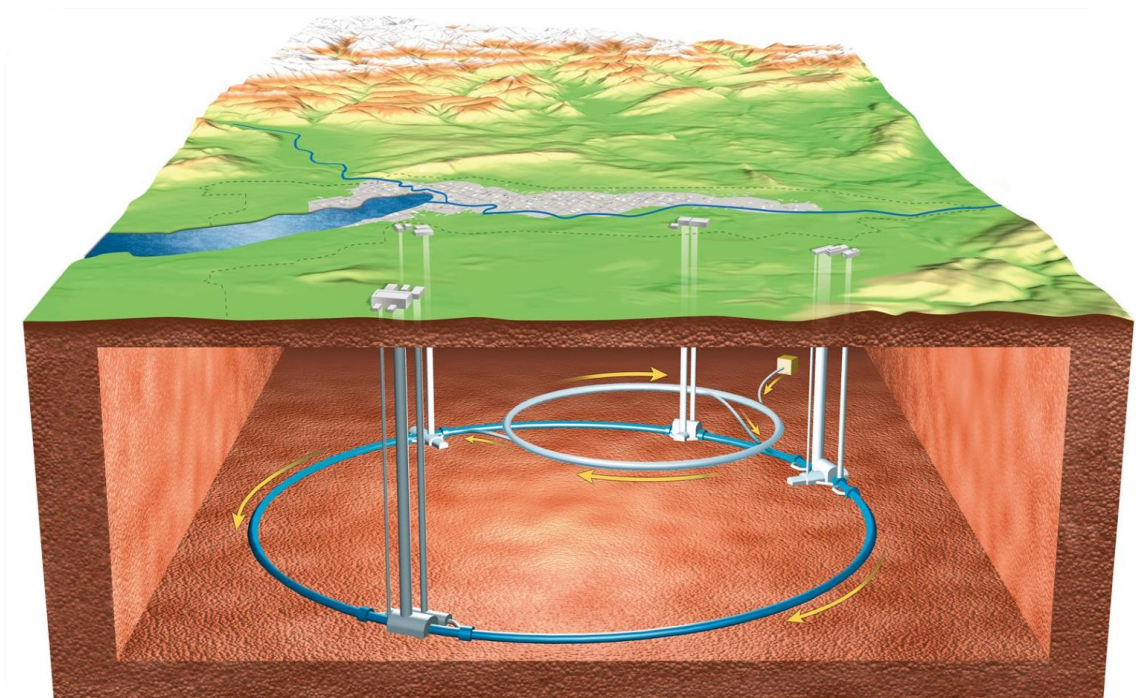


Figura 2.4 - Representação geográfica do complexo de aceleradores

Com as partículas a circular em num tubo de vácuo, o feixe é manipulado através de equipamentos electromagnéticos: **Ímanes-Dipolos** mantêm as partículas em órbita quase circular, **Ímanes-Quadrupolos** focam o feixe, e as **Cavidades de Aceleração** actuam como câmaras de ressonância electromagnética que aceleram o feixe de partículas. Existem cerca de 9600 Ímanes Supercondutores de 15 m de comprimento no LHC, operando a temperaturas mais baixas do que o Espaço sideral ($-270,5^{\circ}\text{C}$) devido às grandes correntes de magnetização da ordem dos 12000 A, conforme apresentado no conjunto de dados da Tabela 2.2.

Tabela 2.2 - Factos Operacionais Gerais do LHC

Quantity	number
Circumference	26 659 m
Dipole operating temperature	1.9 K (-271.3°C)
Number of magnets	9593
Number of main dipoles	1232
Number of main quadrupoles	392
Number of RF cavities	8 per beam
Nominal energy, protons	7 TeV
Nominal energy, ions	2.76 TeV/u (*)
Peak magnetic dipole field	8.33 T
Min. distance between bunches	~ 7 m
Design luminosity	$10^{34} \text{ cm}^{-2} \text{ s}^{-1}$
No. of bunches per proton beam	2808
No. of protons per bunch (at start)	1.1×10^{11}
Number of turns per second	11 245
Number of collisions per second	600 million
(*) Energy per nucleon	

Cada colisão é efectuada no centro de um detector que, tomando partido de um campo magnético e analisando o desvio na trajectória da partícula resultante da colisão é capaz de identificar a sua carga, por exemplo, ou o seu momento de inércia (“*quantidade de movimento*”). Partículas com um elevado momento de inércia viajam em quase linha recta, enquanto as demais realizam espirais apertadas, comportamentos expostos por tipo de detector, conforme visto na Figura 2.5.

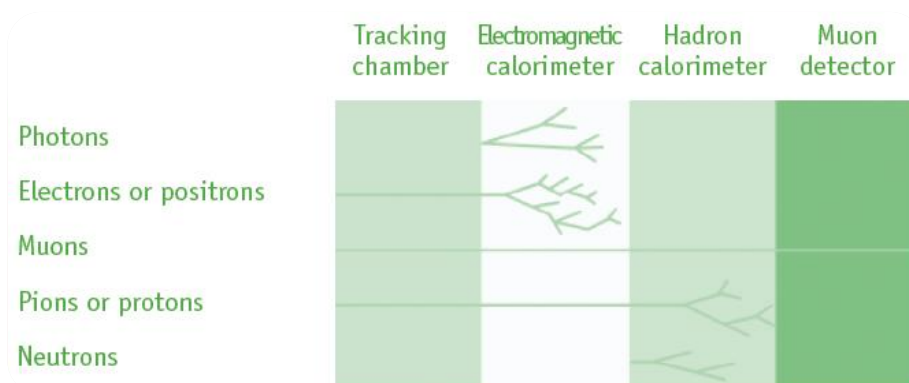


Figura 2.5 - Princípio de funcionamento de Detector por captura de partícula

Existem 6 grandes Experiências independentes no LHC: **ALICE**, **ATLAS**, **CMS**, **LHCb**, **LHCf** e **TOTEM**, com as quatro primeiras a serem instaladas em grandes cavernas subterrâneas nos quatro pontos de colisão do LHC. Cada detector possui a sua tecnologia e propósito característico: ALICE é especializado para colisões de iões de chumbo, observando as propriedades do plasma *quark-gluon*, ATLAS e CMS são detectores multipropósitos destinado a cobrir um grande espectro de actividades desde o Bosão de Higgs à SuperSimetria (*SuperSymmetry* – SUSY), mas de tecnologias e soluções diferentes. O LHCb debruça-se sobre o estudo da leve assimetria entre matéria e antimatéria no universo, e o LHCf uma pequena experiência para o estudo das partículas circundantes ao feixe do LHC para estimar níveis de energia de raios cósmicos ultra energéticos. Finalmente, TOTEM observará as características do feixe mais proximamente.

De facto, encontram-se apresentados na Figura 2.6 uma representação da detecção real do Bosão de Higgs, confirmado em simultâneo e independentemente pelas experiências ATLAS e CMS. A quantidade de dados representada é de uma magnitude de 700 MB/s, ou 15 PB por ano, representando um desafio enorme de execução da rede. De facto, foi precisamente para fazer frente ao desafio de transmissão de dados pelo mundo inteiro que originou a *World Wide Web* precisamente no CERN.

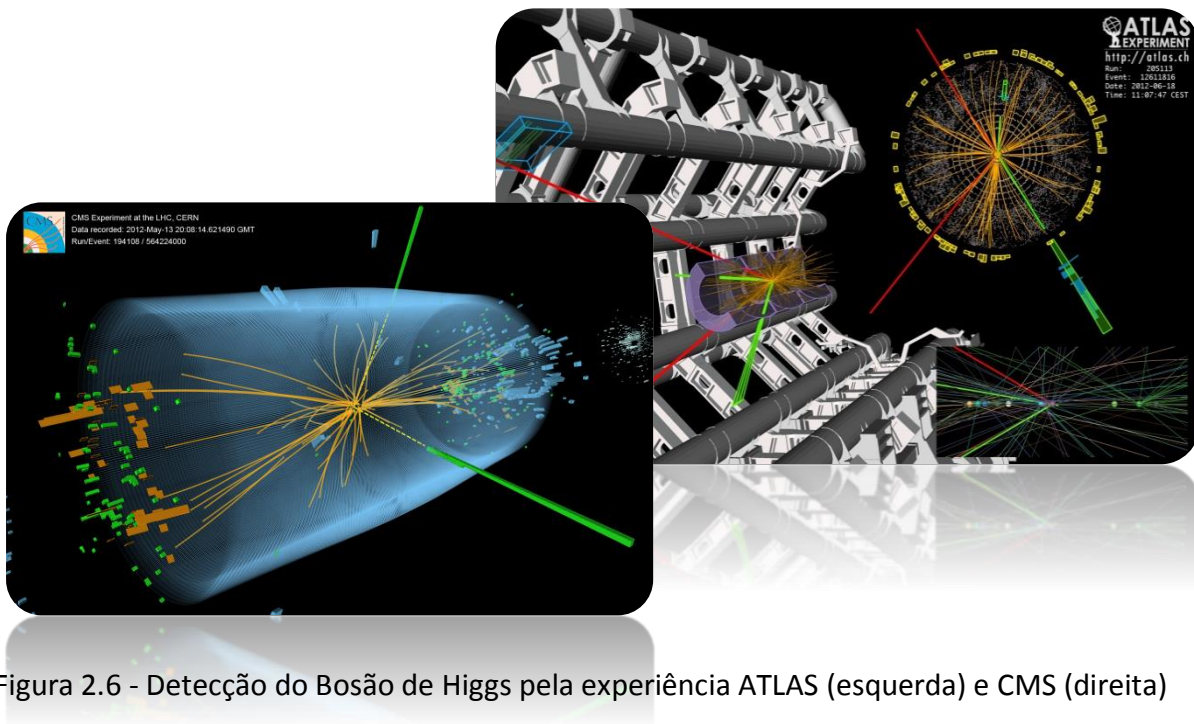


Figura 2.6 - Detecção do Bosão de Higgs pela experiência ATLAS (esquerda) e CMS (direita)

2.3 – Sistema de Coordenadas do CERN e Metodologia

Steve Myers, na altura Director de Operações do LHC aquando da descoberta do Bosão de Higgs em 2012, foi largamente citado na sua afirmação de “fazer com que os dois feixes colidam é comparável a disparar duas agulhas, uma contra a outra dos dois lados do Atlântico e tentar com que colidam a meio caminho”. Num instrumento e experiência desta escala, é crítico que o alinhamento do feixe e de todos os componentes seja o mais preciso quanto possível, implicando precisão a escalas bastante elevadas em desafios de engenharia verdadeiramente únicos. Para tal, um sistema de plano de referência próprio, capaz de representar e descrever todos os objectos presentes no CERN, de equipamentos a edifícios com um elevado nível de precisão era necessário, levando à criação do *CERN Coordinate System* - CCS. Devido a tanto à dimensão da extensão do LHC como dos seus requisitos de precisão, a utilização de uma aproximação da Terra como uma superfície plana ou mesmo esférica como base para este sistema de coordenadas introduz demasiado erro de precisão – conforme explícito no exemplo da Figura 2.7, existe uma clara distinção em Ortometria e Geodesia entre os parâmetros de ‘H’, a altitude ortométrica estabelecida a partir da superfície do nível do mar, e ‘Z’ uma representação vertical local sobre um eixo cartesiano de três dimensões.

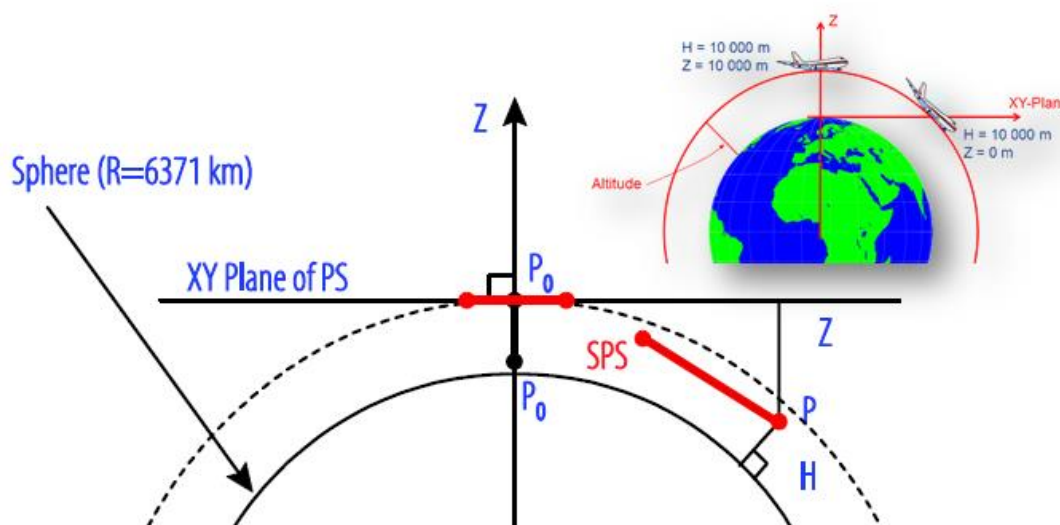


Figura 2.7 – Representação Ortometria: diferença entre ‘Z’ e ‘H’

Para o caso do LHC, um túnel de 27 km de extensão, as aproximações da Terra como ‘plana’ ou ‘esférica’ não são capazes de transmitir o rigor necessário; para o CCS é necessário uma aproximação o mais fiel possível ao real formato da Terra, utilizando-se para tal uma aproximação de uma elipsóide.

Contudo, esta aproximação por si só não é o suficiente, pois falha em traduzir a verdadeira *topografia* da Terra, com bastantes irregularidades (como montanhas), que afectam o campo gravitacional de uma forma localizada.

No caso particular do LHC, dois exemplos de factores geográficos que condicionam largamente este parâmetro são as montanhas Jura a norte e o Lago Léman, a sul.

Para considerar este factor, é necessária uma aproximação a uma **Geóide**, uma **superfície equipotencial gravítica**, onde a orientação do vector gravitacional seja sempre perpendicular [5] [6].

O CCS é então constituído por um **modelo elipsóide local da área, combinado com uma Geóide** para que seja levado em conta este “*desvio da vertical*”. Sendo assim, as coordenadas existentes no CCS referem-se a um conjunto XYZ, referentes à elipsóide, e um parâmetro H, correspondente à altura, mas representando também uma compensação do desvio no campo gravítico conforme a normal da Geóide.

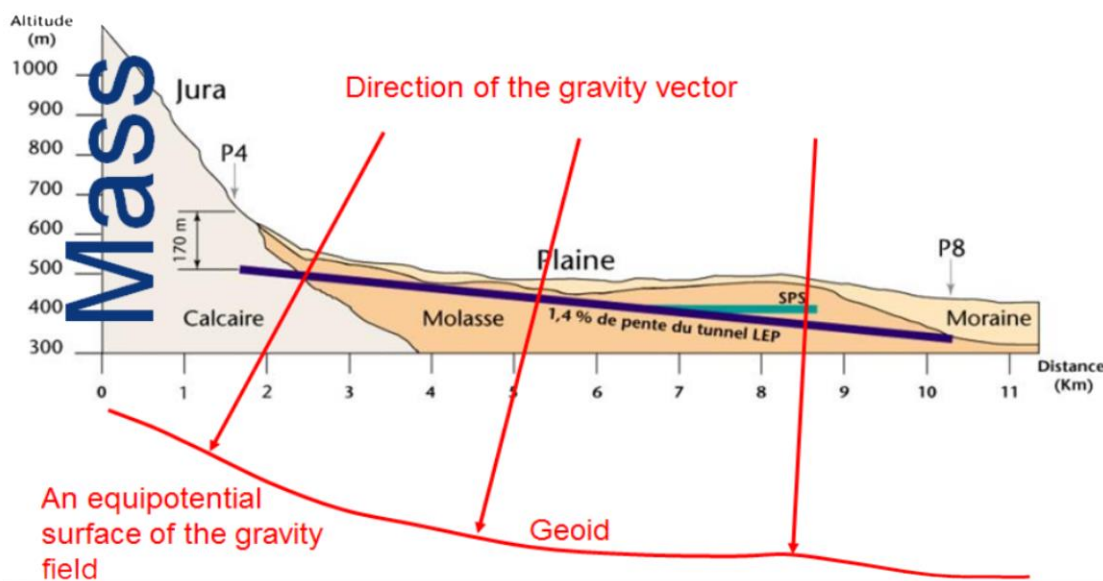


Figura 2.8 – Representação Geóide para o CCS

É precisamente esta necessidade de precisão com o modelo Geodésico empregue pelo CCS, seguindo o modelo exemplificado pela Figura 2.8, que irá condicionar e definir o tipo de tecnologia e metodologia a utilizar no projecto HLS.

O projecto HLS – *Hydrostatic Levelling System* é destinado a **medir a posição e alinhamento dos ímanes supercondutores do LHC**, à semelhança do mesmo representado pela Figura 2.9, que periodicamente devem ser calibrados. A próxima operação de medição será realizada durante o LS1 durante o qual o LHC encontra-se parado para revisões, operações de ampliação das especificações e reparações, e que neste momento prolonga-se até ao início de 2015.



Figura 2.9 – Íman Supercondutor LHC (Dipolo)

Para tal, como forma de medir o desvio e posição vertical entre o alinhamento dos componentes e considerando toda a vizinhança circunstancial do desafio, foi determinado que uma solução contemplando um sistema de **Nivelamento Hidrostático** como a mais indicada, quando contraposto com o enquadramento seguinte.

Como princípio de funcionamento, os sistemas de Nivelamento Hidrostático [7] [8] baseiam-se no princípio de *Vasos Comunicantes*, representado na Figura 2.10 – Exemplo do princípio de Vasos Comunicantes, onde os recipientes com líquido encontram-se conectados de forma comunicativa – seja, que exista possibilidade do líquido fluir de um recipiente para outro livremente, mantendo o nível da superfície do líquido igual entre os mesmos.

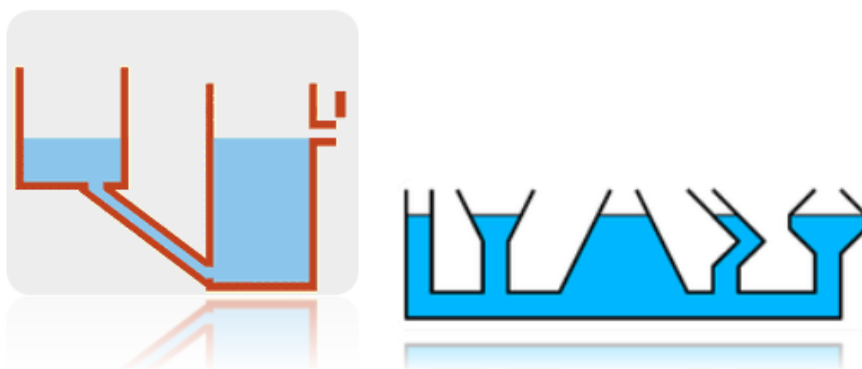


Figura 2.10 – Exemplo do princípio de Vasos Comunicantes

Sendo que o líquido comporta-se como uma superfície equipotencial, com um transdutor ultra-sónico colocado na base, é possível determinar o nível de líquido em cada recipiente. A diferença entre o nível do líquido em ambos os recipientes corresponde precisamente à própria diferença em altura entre os recipientes, e consequentemente, uma relação de referência relativa entre ambos – se a posição absoluta de um for conhecida, é possível determinar a posição dos outros elementos referenciados.

Devido à utilização de uma superfície equipotencial (um líquido como a água), a compensação do desvio gravitacional é permanente – a superfície do líquido estará sempre orientada sobre a normal da Geóide equipotencial gravítica, e desta forma perfeitamente enquadrada com o CCS.

2.4 – Especificações do Sistema

Conforme a descrição e especificações do projecto, é necessário desenvolver um sistema móvel, que seja capaz de medir as diferenças de altura entre os Ímanes Supercondutores do LHC, com uma distância de até 15 m entre os pontos – que, devido à referida inclinação do plano do LHC de cerca de 14 mrad, resulta numa diferença de altura teórica compreendida entre [0, 25] cm, mantendo uma precisão baseada no princípio *As Low As Reasonably Achievable* - ALARA, mas de valor mínimo na ordem dos 50 μm .

Devido ao facto dos sensores hidrostáticos utilizados possuírem alcances de operação típicos na ordem dos 20 mm, longe dos 25 cm passíveis de serem presente entre os dois pontos, é necessário compensar esta diferença mecanicamente através de uma **mesa de translação vertical** e monitorizar este deslocamento de forma coerente ao nível de precisão especificado.

O processo, representado pela Figura 2.11 pode ser contido pela seguinte descrição: as cápsulas contendo os Transdutores ultra-sónicos são colocadas sobre os pontos nos ímanes supercondutores a medir, sendo uma das mesmas capaz de deslocar-se verticalmente, assente sobre uma mesa de translação. Desta forma, mesmo que os sensores estejam colocados fora do seu alcance de operação devido ao nível da água demasiado baixo ou elevado, a cápsula móvel pode então ser deslocada para que esteja ao nível de operação da outra, cuja posição permanece estática.

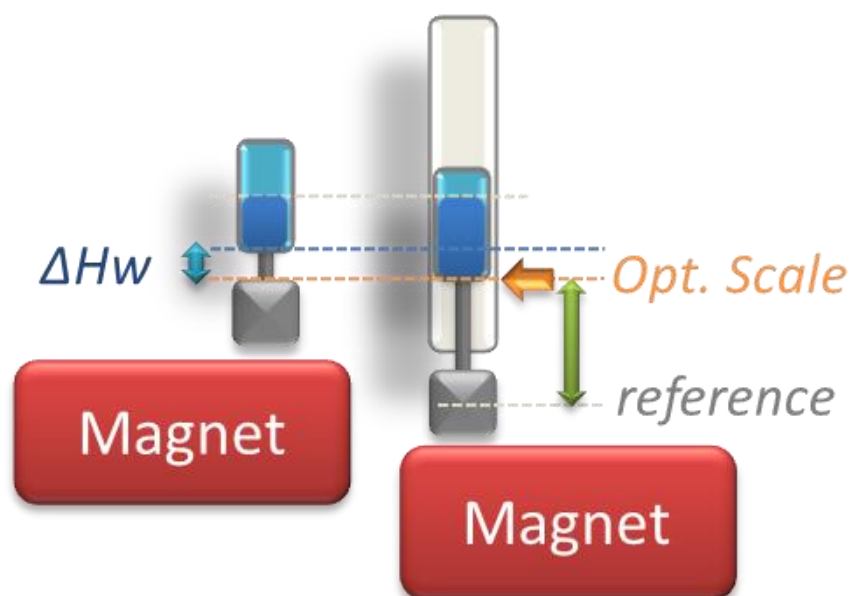


Figura 2.11 – Princípio de operação do sistema HLS

O sistema será assim constituído por **4 elementos de hardware distintos**: um sistema de nivelamento hidroestático utilizando um **sensor ultra-sónico**, (*Ultrasonic Hydrostatic Levelling System – UHLS*), uma **Mesa de Translação**, e um *encoder* digital – **Régua Óptica** de elevada precisão para monitorização do posicionamento. Uma maior descrição da funcionalidade de cada elemento, bem como os parâmetros determinantes da escolha tecnológica do princípio de operação a seguir para o sistema de nivelamento hidroestático são abordados e descritos em maior detalhe seguidamente, nos capítulos pertinentes.

A aplicação estará baseada numa plataforma PXI da National Instruments, com todos os *drivers* e lógica de controlo a serem desenvolvidos utilizando o *software* LabVIEW.

Na Figura 2.12 encontramos representados individualmente todos os diferentes elementos de *hardware* que constituem o sistema:



Figura 2.12 – Componentes de *hardware* principais do HLS

Onde:

- 1 – Cápsula 1 (móvel) montada no sistema da mesa de translação Zaber, com Régua Óptica integrada
- 2 – Cápsula 0 (estática), com a peça de referência visível no topo, e o sensor ultra-sónico na base
- 3 – PXI (Windows 7 OS)
- 4 – *UHLS Metering Point Electronics – MPE*, electrónica constituinte do módulo.



Figura 2.13 Componentes *Hardware* HLS. A azul, o Módulo *UHLS*, a amarelo o *Encoder* Óptico e a vermelho o Motor a Passo e mesa de translação

Todo o Sistema de Controlo, incorporando os drivers para os diferentes equipamentos cuja integração encontra-se representada na Figura 2.13, será construído em LabVIEW e posteriormente colocado numa plataforma PXI da National Instruments, um *standard* da instrumentação industrial, com o sistema operativo Windows. Esta plataforma de desenvolvimento em linguagem gráfica “G”, utilizando LabVIEW oferece uma grande velocidade de desenvolvimento, sendo uma poderosa ferramenta de desenvolvimento e especialmente indicada como *middleware* devido à facilidade de interligação com bibliotecas e *dlls* de diferentes formatos, desenvolvidas em outras plataformas, resultando numa excelente capacidade de comunicação e interoperabilidade com um grande número de equipamentos ao mesmo tempo que suporta uma interface gráfica avançada.

Conforme as especificações e cronograma do projecto, é necessário o desenvolvimento do projecto dentro da janela temporal do LS1. Embora seja um dos requisitos que o sistema de HLS seja móvel, por forma a efectuar medições sobre todos os Ímanes ao longo LHC, nesta fase de aplicação a sua operação será manual, significando que existirá uma dependência a um operador que colocará e operará o sistema em cada ponto de medição pretendido.

Futuramente é previsto que o projecto *Mobile-HLS* integre parte do *Survey-Train*, um comboio montado num único carril que percorre ao longo do LHC, conforme visto na representação da Figura 2.14 e que seja autónomo ao ponto de realizar as medições de maneira remota, evitando também a presença de operadores em áreas restritas, e possivelmente radioactivas durante períodos de operação do LHC [9].

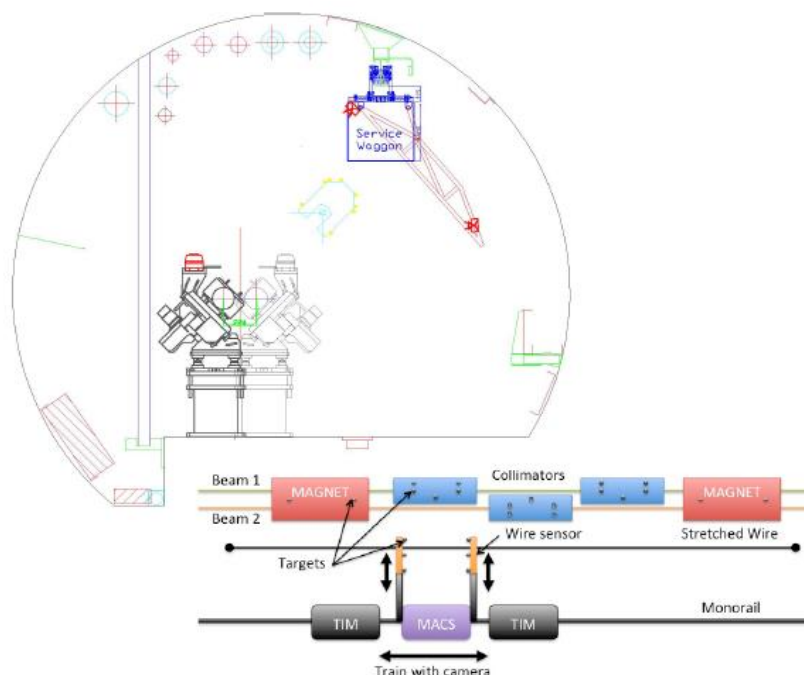


Figura 2.14 – Representação do *Survey Train*

Desta forma podemos sumarizar as principais directivas do projecto da seguinte forma:

- Sistema capaz de **medir diferenças de alturas entre dois pontos**, situados até **15m entre si**, e dentro de um **alcance de operação de [0; 25] cm**, com uma **precisão mínima de 50 μm** ;
- Deve manter a **mobilidade e portabilidade** necessária para possibilitar operação através de um operador nos túneis do LHC, e posterior integração no *Survey Train*.

2.4.1 – Sistema de Nivelamento Hidrostático Ultra-Sónico

Os sistemas de Nivelamento Hidrostáticos são frequentemente utilizados para monitorizar a estabilidade ou desvios verticais com elevada precisão, ao nível do micron. Existem diversas técnicas de aplicação, baseadas no princípio de vasos comunicantes já referido, caracterizadas pela técnica de medição do nível do líquido: **Tácteis**, **Ultra-sónicos** ou **Capacitivos**. Embora robustos, os sistemas do tipo Táctil dependem de uma agulha que determina a posição do líquido ao efectuar contacto com o mesmo, o que devido à elasticidade da superfície da água, levanta questões da validade da medição, especialmente se contemplando a repetição de medições onde o líquido deposita-se na agulha e a gota resultante compromete a medida.

Face ao mesmo progredimos para os dois tipos de tecnologias restantes, e são os de tipo ultra-sónico que possuem uma vantagem determinante face aos do tipo capacitivo, possuindo tanto a sua parte activa – o transdutor, como uma peça de referência *imersos* na água. Sendo que a superfície do sensor capacitivo é muito sensível à presença de água na superfície do eléctrodo, tal solução não é ideal para uma aplicação móvel, onde o contacto com o líquido é praticamente inevitável; a degradação passível de ser daí resultante é perfeitamente visível na Figura 2.15 e determinante para a seleção do tipo de tecnologia dos transdutores a utilizar.



O Módulo *UHLS*, um sistema destinado para medição da diferença de altura entre os dois pontos, apresentado singularmente na Figura 2.16, é um protótipo especialmente desenhado e concebido com os requisitos deste projecto em mente, desenvolvido pelo Centro de Pesquisa e Laboratórios *Deutsches Elektronen-Synchrotron* – DESY [10].

O sistema integra dois transdutores Piezoeléctricos Ultra-sónicos, conectados através de um cabo coaxial de 50Ω, representados pela indicação de PT – *Piezoelectric Transducer* no diagrama da figura Figura 2.17. O pulso ultra-sónico gerado por estes é originado a partir de um pulso eléctrico cujas características de amplitude e duração são contidas nos intervalos de [120, 160] V e [20, 60] ns, respectivamente.

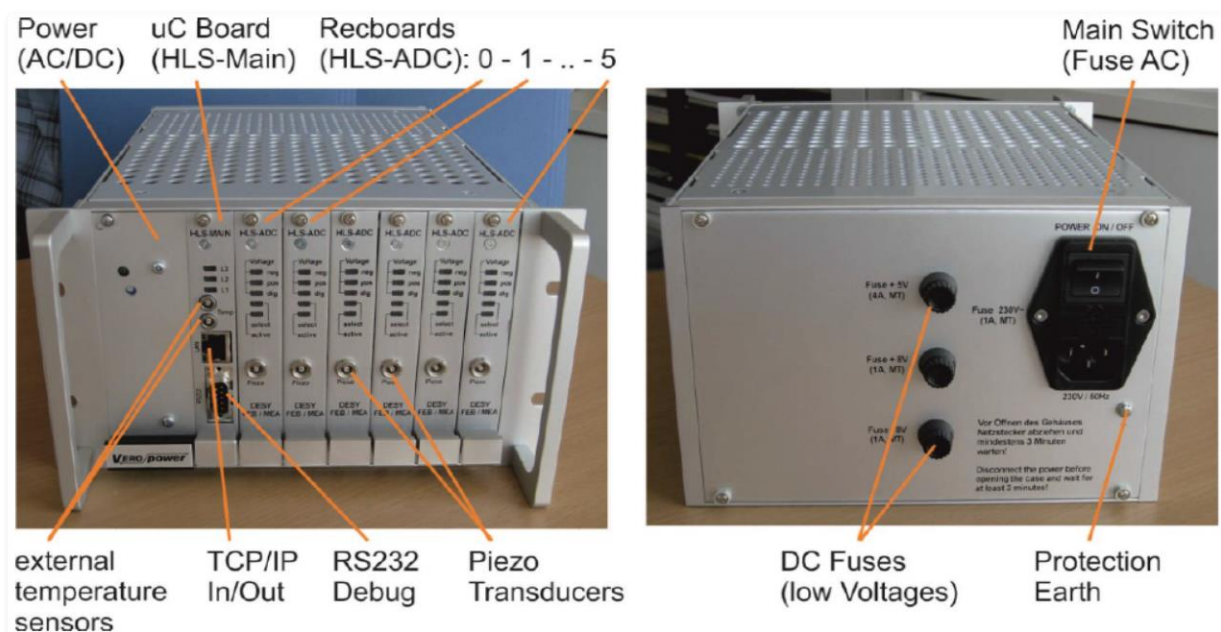


Figura 2.16 – Módulo sensor *Ultrasonic Hydrostatic Levelling System*

Cada *Metering Point Electronics* – MPE, a designação dada à electrónica de aquisição e tratamento do sinal proveniente do sensor, utiliza o protocolo de comunicação estabelecido via TCP/IP. Um microcontrolador embebido analisa os comandos enviados e realiza os endereçamentos necessários dos mesmos através do *backplane bus* para o módulo HLS-ADC transdutor correspondente, referido como *RecBoard* pela documentação correspondente [10]. Existem 5 *RecBoards* disponíveis, conforme ilustrado no diagrama representado na Figura 2.17.

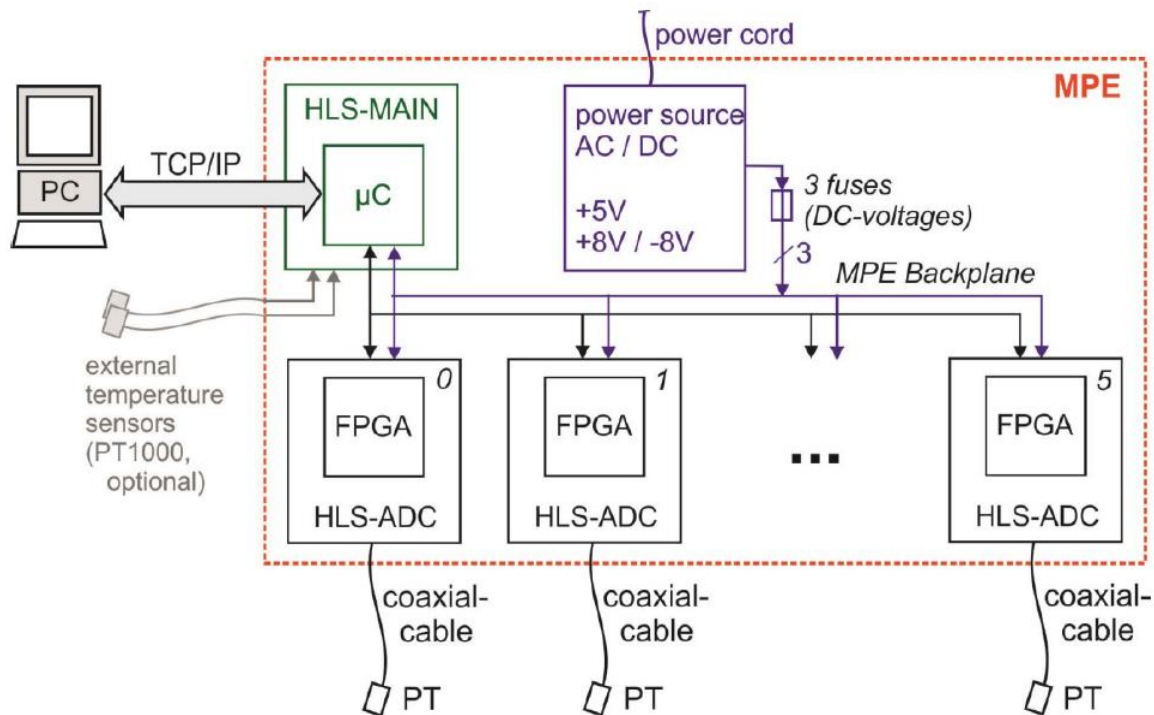


Figura 2.17 – Esquemático da electrónica de funcionamento da MPE

O seu princípio de operação baseia-se nos transdutores piezoeléctricos utilizando a técnica *impulso-eco*, gerando um sinal de pulso ultra-sónico e medindo as reflexões do mesmo sobre a superfície da água e uma peça de calibração de referência. Através deste método, conhecendo previamente as dimensões da peça de referência, é possível realizar uma estimativa da velocidade de propagação do som e consequentemente da distância percorrida, sem obrigatoriamente levar em conta a temperatura e pressão do líquido – tornando o sensor efectivamente *autocalibrante*.

Conhecendo o nível de água em um dos recipientes (cápsulas), devido ao líquido ser mantido num estado comunicante, a sua superfície estará nivelada ao mesmo nível equipotencial, conforme exposto anteriormente. Se a posição em altura de uma cápsula for então conhecida é possível calcular com precisão a posição do outro recipiente, relativa ao primeiro.

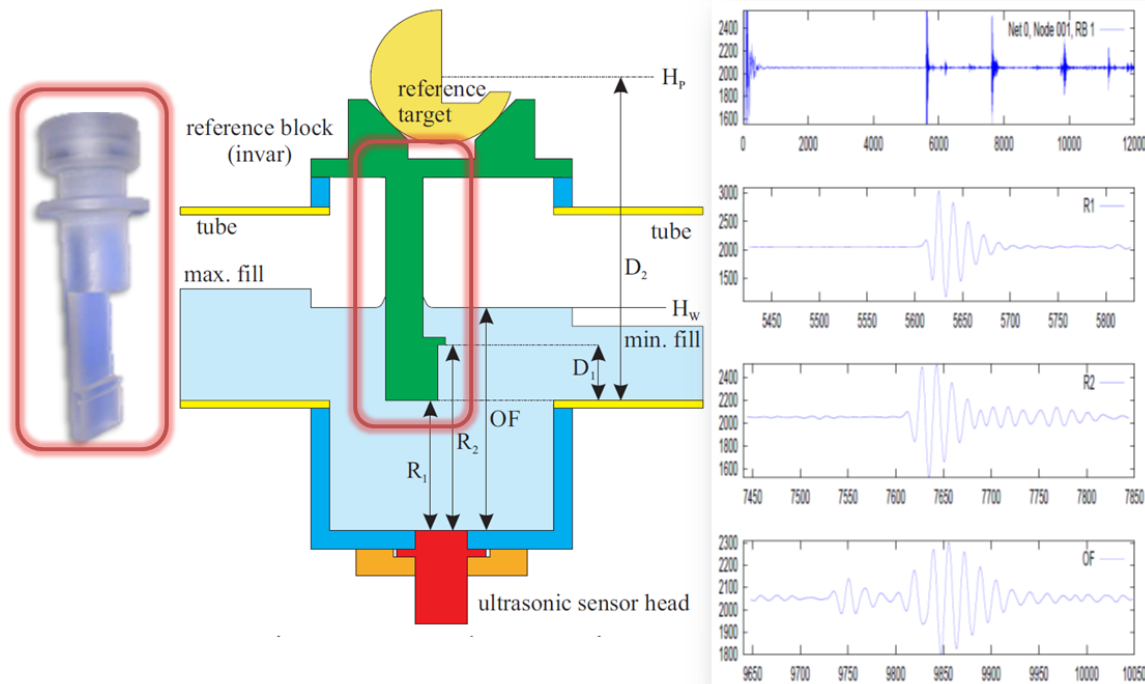


Figura 2.18 - Peça de Referência em Quartzo (esquerda), esquemático de reflexões (centro) e análise das reflexões da técnica *Impulso-Eco* (direita) [7] [10]

Na Figura 2.18 encontra-se representado o princípio de funcionamento da técnica *Impulso-Eco* anteriormente referido onde é possível visualizar as reflexões obtidas a partir do pulso e a sua curva característica. A partir da análise da mesma é possível identificar se a reflexão é proveniente da peça de referência ou da superfície da água, com “zero-crossing” de sinal negativo e positivo, respectivamente.

A identificação dos factores da expressão (2.1) segue a denominação expressa pela Figura 2.19. Desta forma, o nível da água H_w é dado por:

$$H_w = D1 * \frac{OF - R_1}{R_2 - R_1} \quad (2.1)$$

Onde:

- **H_w**: Nível da água no recipiente;
- **D1**: parâmetro de calibração da peça de referência, distância previamente conhecido;
- **R1** e **R2**: distâncias entre os limites da peça de referência, necessários por forma a permitir a calibração com D1
- **OF**: tempo de propagação da reflexão com a superfície da água.

Sendo que a velocidade de propagação do pulso ultra-sónico é fortemente dependente de factores como a temperatura, e logo da densidade da água, a elegância do conceito do sistema reside na remoção deste parâmetro como factor. Com a distância D1 previamente conhecida, a distância até OF, representando o nível da água, pode ser calculada pela expressão (2.1), onde os factores da velocidade presentes em todos os termos anulam-se, efectivamente tornando os factores R1, R2 e OF representações de uma relação indistinta entre distância e tempo, ocorrendo muitas vezes sendo referida coloquialmente como ‘reflecção’. Este conceito entre as duas grandezas é de tal forma intrínseco ao sistema que encontra-se presente em produções elaboradas pelos criadores do sistema; na referência [8] podemos encontrar OF descrito como uma distância, e em [10] o mesmo assume um manto de designação como “tempo”.

Seguidamente na Figura 2.20 é possível visualizar um exemplo de dados brutos não tratados provenientes do ADC do transdutor onde é possível distinguir o pulso inicial das suas reflecções resultantes, R1, R2 e OF.

A qualidade do sinal recebido e das três reflecções, os problemas que surgiram e a sua resolução e finalmente todos os parâmetros de configurações definidos serão descritos mais aprofundadamente no subcapítulo 3.9 – Sensor de Nivelamento Hidrostático Ultra-sónico (UHLS) descrevendo a implementação do módulo, aquando do desenvolvimento do seu sistema de Controlo em LabVIEW.

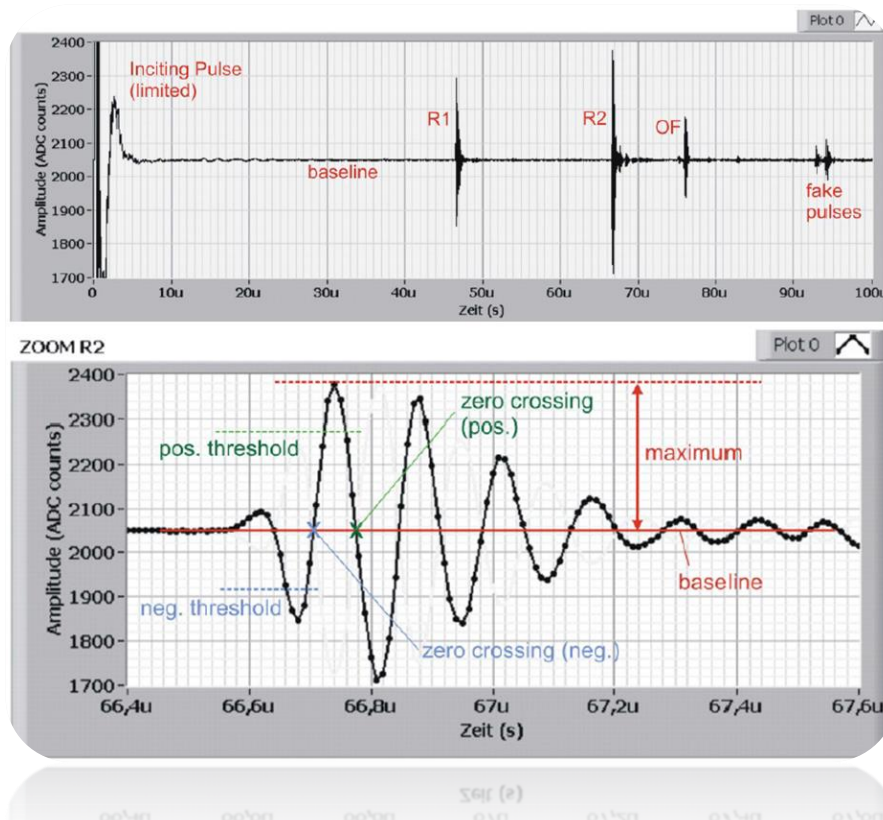



Figura 2.20 – Acima: Reflecções em dados brutos. Abaixo: pormenor da reflecção R2

2.4.2 – Mesa de Translação: Zaber Dovetail

Conforme referido, como os sensores ultra-sónicos em cada cápsula apenas possuem um alcance efectivo para determinar o nível de água na ordem dos 25mm, e como as diferenças de altura entre os pontos podem variar de 0 a 25cm, é necessário deslocar um dos recipientes para que fique próximo do nível do outro, colocando os sensores dentro do seu alcance operacional.

Para efectuar esta deslocação foi utilizado uma mesa de translação Zaber Dovetail T-LLS-260. Permitindo até 260mm de deslocação, com um Motor a Passo de resolução 0,16 μm e sensor de posição “Home” tipo Hall integrado para detecção da posição inicial, necessita apenas de uma alimentação de 15V com controlo efectuado por computador. A documentação disponibilizada pelo fabricante, expondo tanto as especificações do equipamento como apresentando um guia de utilização, encontra-se disponível em: <http://www.zaber.com/>, sendo digna de nota pela sua qualidade de redacção e apoio e exemplificada pela Figura 2.21.



Specification (click for definition)	Value
Microstep Size (Default Resolution)	0.15625 μm
Integrated Controller	Yes
Travel Range	260 mm
Accuracy (unidirectional)	40 μm
Repeatability	< 3 μm
Backlash	< 15 μm
Maximum Speed	6 mm/s
Minimum Speed	.001465 mm/s
Speed Resolution	.001465 mm/s
Communication Interface	RS-232

Figura 2.21 – Mesa de translação Zaber Dovetail e suas principais características

A comunicação com o actuador será estabelecida via RS232, com as definições de comunicação de 9600 Baud rate, sem *hand-shaking* e 1 Byte de dados sem paridade. Todas as instruções são dadas por um conjunto de 6 Bytes, conforme a seguinte estrutura:

- Byte 1: Número do Dispositivo
- Byte 2: Número do Comando
- Dados (LSB)
- Dados
- Dados
- Dados (MSB)

Uma aplicação de controlo própria do fabricante encontra-se disponível e foi utilizada inicialmente para propósitos de testes e *trouble shooting*, sendo todo o desenvolvimento do *driver* respectivo para a aplicação feito em LabVIEW e abordado no capítulo referente à implementação e desenvolvimento do sistema.

2.4.3 – Encoder Linear: Régua Óptica Renishaw

Para a monitorização do deslocamento, devido ao elevado grau de precisão requerido não é possível depender do *encoder* óptico integrado na própria mesa de translação onde o motor a passo é assente, de resolução muito inferior à pretendida e propenso a “saltos” e integração de erros consoante a operação. Desta forma, para conhecermos a posição do sensor móvel, é utilizado uma escala/régua óptica de elevada precisão, modelo *Renishaw RSLA* configurado com uma resolução de 0,15 μm . Apresentado na Figura 2.23 pode ser visualizado uma representação do seu diagrama de operação. Esta encontra-se montada na vertical, com o leitor apresentado na Figura 2.22 fixo à cápsula, e conforme visto em maior detalhe anteriormente na Figura 2.13

A interligação física com o sistema (PXI) é realizada através de um adaptador BiSS-USB, modelo MB3U, fabricado pela empresa **icHaus**.

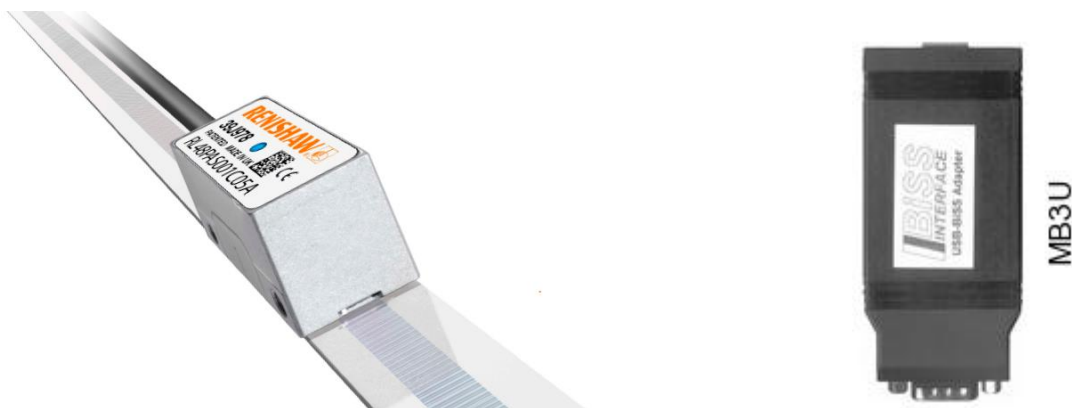


Figura 2.22 – Régua Óptica *Renishaw Optical Scale* e Adaptador BiSS-USB

O protocolo de comunicação utilizado é o *Bidirectional Serial Synchronous* - BiSS, uma interface digital *open-source* compatível com o *standard* industrial *Serial Synchronous Interface* - SSI, mas oferecendo também funcionalidades adicionais:

- Comunicação Serie Síncrona
- 2 Linhas unidireccionais (*Clock*, *Data*)
- Compensação de atraso de linha para transferências de alta velocidade
- Sistemas de segurança de integridade de dados: CRC, Erros e Avisos
- Capacidade de actuação como Bus, mantendo diversos equipamentos numa linha

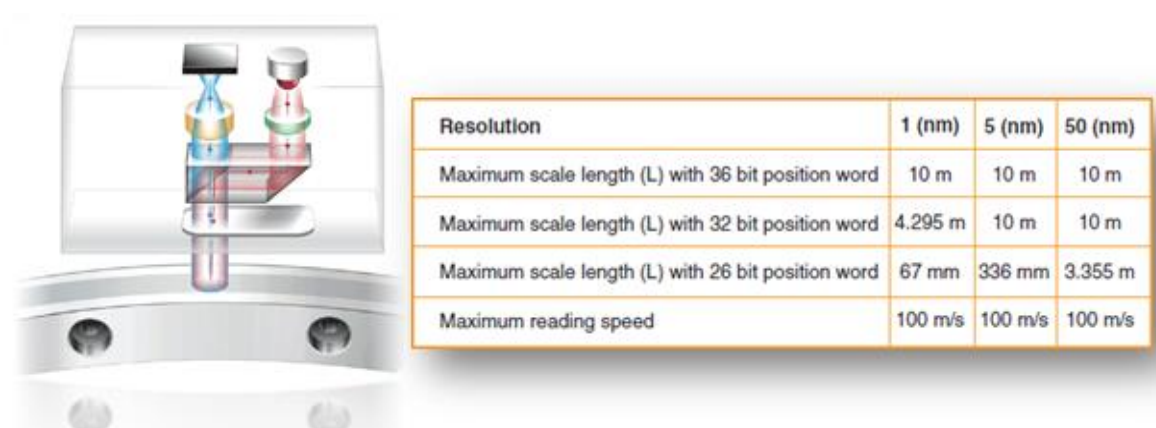


Figura 2.23 – Diagrama de Funcionamento da Régua Óptica e suas principais características

Este protocolo possibilita a operação em modo de comunicação bidireccional contínua (*C-Mode*), permitindo a comunicação de parâmetros de forma bidireccional sem que a transmissão de dados do sensor ou que as instruções de *Master* para *Slave* sejam interrompidas - um aspecto crítico em operações de controlo digital rápido, como é o caso para controladores de motores com *feedback loop*. Conforme visto em maior detalhe na Figura 2.24, cada *Slave* opera sob um identificador único, mantido para cada ciclo de operação. Cada fim de ciclo é interpretado como um *timeout*, e utiliza o último nível lógico com o bit de controlo para os *Slaves* conectados, que então compilam os seus telegramas de dados através de vários destes ciclos isócronos.

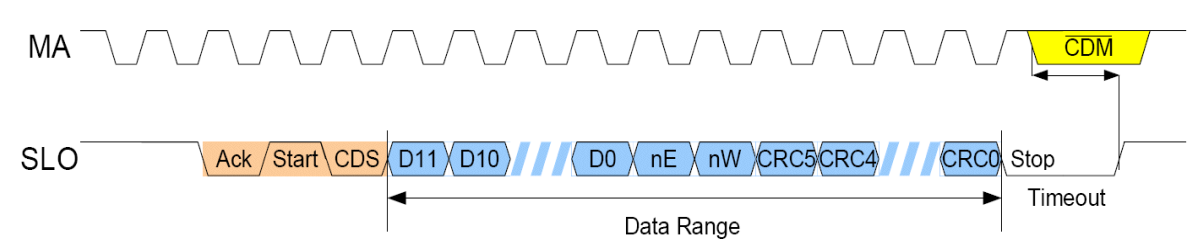


Figura 2.24 – Telegrama de mensagem Protocolo BiSS

CAPÍTULO 3 – PROJECTO HLS

3.1 – Desenvolvimento & Implementação

O cronograma de desenvolvimento da aplicação, apresentado na Figura 3.1, estende-se por toda a duração do estágio, embora outras tarefas menores tenham sido executadas em paralelo e fora do contexto do projecto. Exposto na Figura 3.1, encontra-se um diagrama temporal representando a ordem e principais marcos de desenvolvimento. Note-se que este é meramente indicativo da ordem de acontecimentos, não da sua duração relativa.



Figura 3.1 – Cronograma de desenvolvimento do projecto HLS

A fase representada como *Testes de Operação* de facto contempla a realização dos testes no seu ambiente final, dentro do LHC.

O processo de “*Debugging* e Novos Requisitos” enquadra os resultados da primeira fase de testes num enquadramento de processo iterativo, permitindo que estes exerçam as suas inferências, debruçadas sobre o funcionamento actual do sistema e permitam discernir requisitos que não se encontravam até então definidos nas especificações base do projecto, provenientes de uma base de operação prática.

Além do desenvolvimento do projecto HLS, durante os meses de Agosto e Setembro foi também prestado suporte a dois outros projectos externos à secção, denominados de ‘COLDEX’ e ‘Wire Scanner’. Ambos os projectos possuem características semelhantes, como a necessidade de interligação de diferentes tipos de *hardware*, um dos quais sendo um motor ou peça móvel, através de um sistema de controlo único, completo com registo e análise de dados. Embora fora do âmbito directo deste relatório, a documentação associada está disponível mediante pedido.

3.2 – Ambiente de Desenvolvimento

O desenvolvimento de todo o sistema de controlo foi realizado utilizando o *software* LabVIEW 2013 com uma posterior migração para a versão 2014, com o *add-on* da *palette de Rapid Application Development Environment* – RADE. Esta ferramenta, embora *open-source*, foi criada e continua a ser desenvolvida pelo CERN como forma de estabelecer uma solução de alto nível para o desenvolvimento e interligação de aplicações em diferentes sistemas, combinando bibliotecas de diferentes tipos como MatLab, Java ou WinCC através de LabVIEW. Os seus principais princípios de aplicação, apresentados na Figura 3.2 podem ser descritos como:

- **Aplicações de tempo de desenvolvimento curto**, (i.e. Análises específicas)
- **Aplicações de rápida evolução** (i.e. Aplicações em sistemas de *hardware*)
- **Aplicações de uso não recorrente** (i.e. ferramentas de diagnóstico técnico específicas)
- **Sistemas *Stand-Alone*** (i.e. Sistemas de ambientes de testes e simulação)

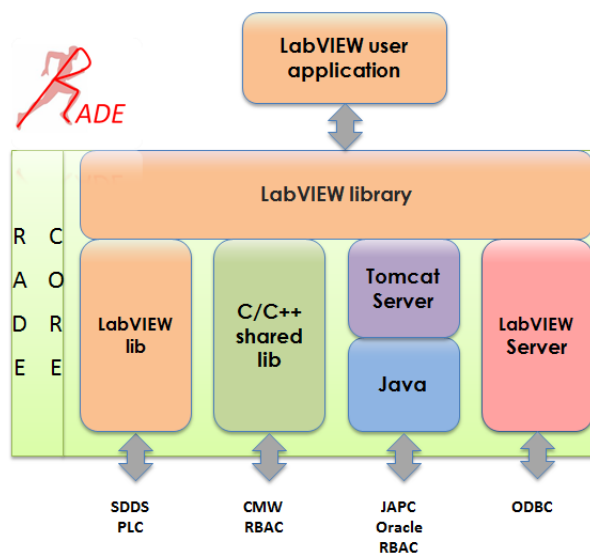


Figura 3.2 – Diagrama de funcionamento da ferramenta RADE

Esta contém assim uma colectânea de bibliotecas e *Virtual Instruments* – VI, essencialmente o termo designado para blocos encapsuladores de funções em LabVIEW, desenvolvidos pelo CERN ou pela comunidade de LabVIEW em geral - na sua essência, uma adição ao *framework* indicada para o desenvolvimento de ferramentas de alto nível, e encontra-se disponível em:

<https://j2eeps.cern.ch/wikis/display/EN/RADE>

O ambiente de desenvolvimento também contempla a utilização da ferramenta de gestão de desenvolvimento de *software* – **Subversion SVN**, em combinação com a aplicação de SVN Client - **Tortoise**, sendo desenvolvido constantemente dentro de um repositório interno ao CERN.

Outros requisitos que devem estar presentes para a reprodução correcta do ambiente de desenvolvimento e execução incluem os ficheiros de configuração *.ini* de cada módulo da aplicação, embora tal não seja crítico devido a existir uma redundância com os mesmos presente no código da aplicação que garanta a execução, durante a qual o operador pode definir e guardar um novo conjunto de ficheiros *.ini*.

3.3 – Projecto LabVIEW

Tanto os controladores de *hardware* como a infra-estrutura da aplicação em si, integrados na Lógica e Sistema de Controlo da aplicação foram desenvolvidos utilizando LabVIEW.

Este capítulo irá abordar o desenvolvimento do projecto HLS segmentado pelos seus Módulos constituintes, seguindo uma ordem que não corresponde necessariamente à ordem cronológica de desenvolvimento, mas por forma a manter uma sequência lógica de interligação referente às suas funcionalidades. Do mesmo modo, embora a descrição de cada módulo será mantida o mais elementar possível, devido à estrutura distribuída da aplicação a referência a módulos circundantes é inevitável.

A metodologia de desenvolvimento de *software* procurou seguir os princípios de *Agile & Scrum programming*, sendo cada funcionalidade do sistema testada independentemente.

De acordo com as melhores práticas de LabVIEW, todo o código foi mantido o mais modular possível, fabricando as camadas de abstracção necessárias para uma leitura fluida de toda a aplicação.

3.4 – HLS_Main e a Arquitectura da Aplicação

Os paradigmas seguidos para o desenvolvimento da arquitectura da aplicação visam manter cada módulo o mais independente e *stand-alone* possível (visando *unit-testing*), sendo lançados de forma assíncrona e correndo em paralelo essencialmente como um processo *daemon*. Os módulos utilizam também uma estrutura baseada num sistema de comunicação avançado convencionalmente reconhecido como um *Queue based Message System*, com um mecanismo de operação interno utilizando Eventos. Desta forma, utilizando estas ferramentas de Eventos como *triggers*, comunicação via *Queues* e processamento paralelo das diferentes instâncias obtemos uma grande optimização de performance, evitando os pontos de engarrafamento (*bottlenecks*) que comprometem seriamente a execução da aplicação e que ocorrem vulgarmente em sequências lineares.

Sobre a lógica de controlo, os módulos são mantidos o mais independentes possível, com uma medida de inteligência distribuída, mas de forma geral são limitados na sua capacidade de tomar decisões (nomeadamente relativas ao próximo estado de execução a seguir), com a excepção dos processos de terminação e limpeza dos recursos alocados. Exposto de outra maneira, de regra geral um módulo terá algum grau de liberdade e autonomia da aplicação central, com inteligência suficiente para determinar ou alterar o melhor rumo de acção para uma dada instrução recebida (de visão mais ‘introspectiva’), mas actuando em resposta e de acordo com comando recebido – sem a autoridade de tomar grandes decisões de *navegação de estado*, com a excepção da sua própria terminação no caso de determinar algum comportamento anormal da função *Main*.



Figura 3.3 – Representação de Funcionamento da função *Main*

O diagrama básico de operação da *Main* encontra-se representado anteriormente na Figura 3.3, enquanto na Figura 3.4 enquadra-se uma representação em maior detalhe da *Graphical User Interface* - GUI da **Main_HLS.vi**. Sendo intrinsecamente uma aplicação complexa, procurou-se utilizar as melhores práticas de *Human Machine Interface* - HMI para estabelecer uma GUI o mais intuitiva e compartimentada possível, por exemplo, discriminando os controlos e *displays* de dados conforme a sua natureza e funcionalidades.

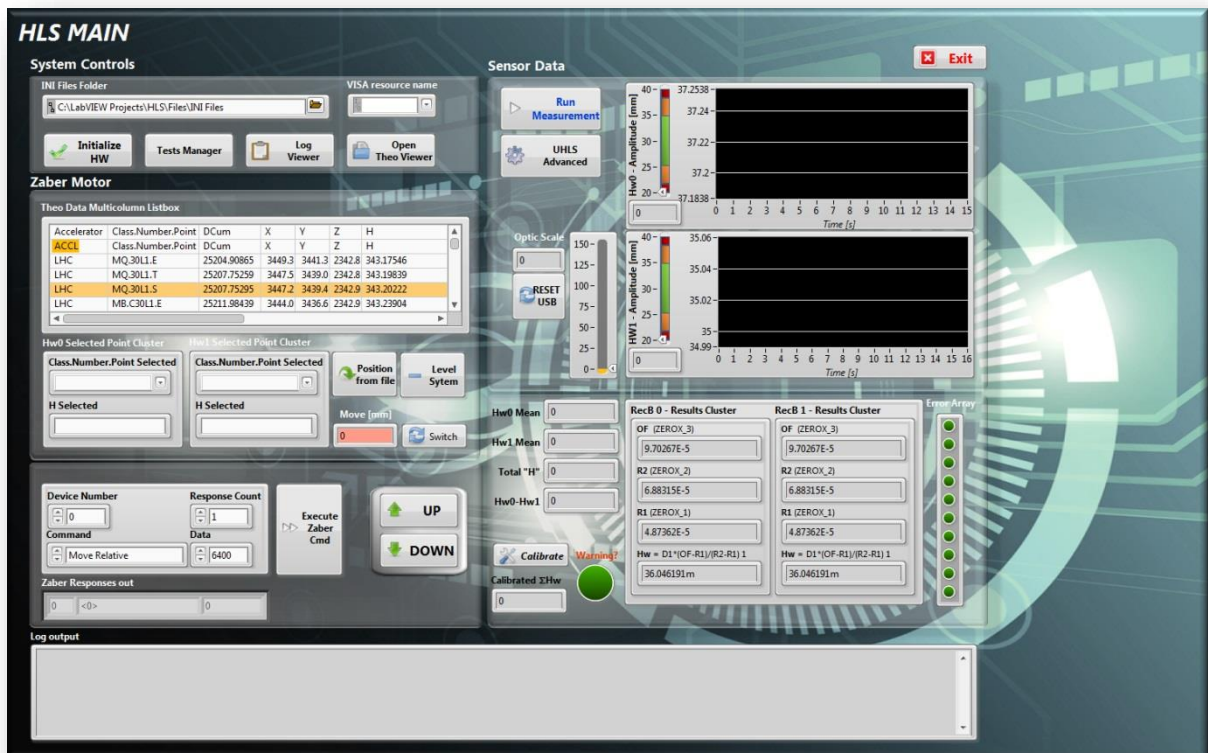


Figura 3.4 – GUI da *Main*, versão completa

A **Main_HLS.vi** actua como o *hub* central do sistema de controlo, com duas grandes funções principais, contidas respectivamente nos dois ciclos de operação, apresentados em maior detalhe na Figura 3.5. A primeira é responder aos *User Events*, eventos criados pelo Operador, interpretando os pedidos, gerindo e enviando os comandos apropriados através dos canais de comunicação adequado – as *Queues*. A segunda é apresentar a informação adequada ao operador, proveniente de diversos pontos da aplicação. A representação destas funções pode ser vista seguidamente na mesma figura, com os principais VIs contidos em cada *loop* representados pelos seus ícones. As *Queues*, que permitem a comunicação entre diferentes ciclos, são representadas pelas “setas” que atravessam os limites dos mesmos.

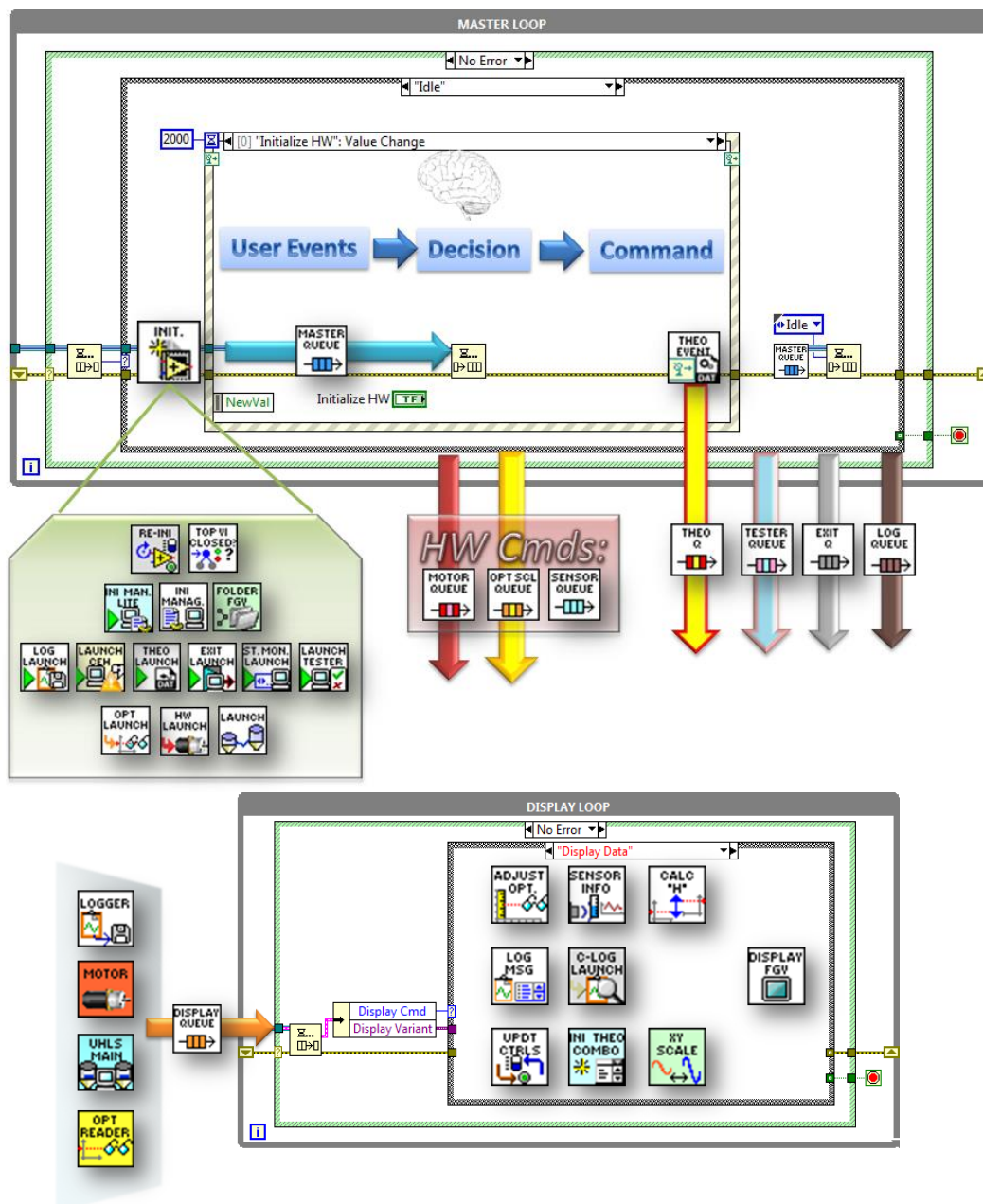


Figura 3.5 – Representação da função Main e dos seus canais de comunicação (*queues*)

Na representação exposta, visando o denominado “*Master Loop*” podemos identificar a estrutura básica do ciclo e os principais *sub-VIs*, cujo comportamento de operação é o de uma máquina de estados movida por uma *queue* que é actuada pelos eventos do operador. Quando o utilizador “dispara” um evento, este é captado e interpretado, reagindo proactivamente por forma a encaminhar as instruções apropriadas para os Módulos respectivos através das *Queues* respectivas, representadas na Figura 3.5 pelas ‘setas’ que atravessam as fronteiras do ciclo de execução.

Exceptuando algum condicionamento e formação de Dados enviados para o “*Display Loop*”, a *Main* efectua muito pouco processamento dos eventos, ‘*outsourcing*’ as grandes operações de processamento para os outros módulos correndo em paralelo. A excepção a

tal são os processos de inicialização dos próprios módulos, descritos seguintemente no subcapítulo 3.4.3.1 – INI File Manager.

3.4.1 – Estratégia de Tratamento de Erros

Outro aspecto crítico da arquitectura é a questão de tratamento de erros. Com cada um dos módulos principais contidos num ciclo, na sua maioria actuando de forma já referida como uma Máquina de Estados movida por uma *Queue*, onde cada subfunção constituinte, denominadas de *Virtual Instrument* - VI em LabVIEW, é encapsulada por uma *Error Case Structure* por forma a que, caso ocorra um erro, este seja passado até ao final da actual iteração do ciclo, conforme visto no exemplo apresentado na Figura 3.6. Sendo que este é mantido num *shift register*, a próxima iteração do ciclo encaminhará o erro para a entrada no *Error Case* definido, segundo a representação vista na Figura 3.6.

Os mecanismos deste processo são vistos em maior detalhe no capítulo 3.10 – Tratamento de Erros, com a estratégia apenas referida neste ponto como forma de unir este conceito com o conceito arquitectural.

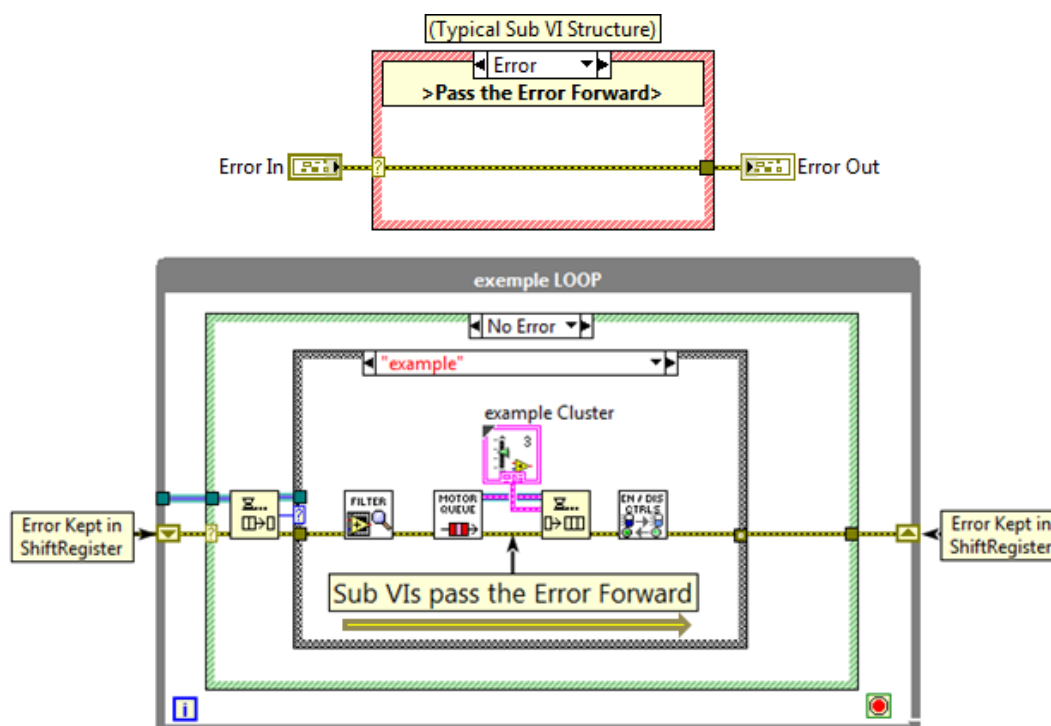


Figura 3.6 – Exemplo dos mecanismos de encaminhamento para tratamento de erros embutidos na infra-estrutura da aplicação

3.4.2 – Main_Lite

Prevendo uma expansão da mobilidade operacional do sistema HLS, uma especificação adicional emergiu, como forma de permitir o controlo remoto da aplicação a partir de uma fonte de controlo externa ao PXI que contém a execução da aplicação. Assim sendo, podemos decompor esta especificação em dois componentes: **Publicação via Web** da aplicação por forma a permitir acesso remoto proveniente de uma rede ou dispositivo específico através de um *browser* com apenas o *plugin LabVIEW Runtime Engine* como requisito, e o desenvolvimento de uma **versão para uso com um dispositivo tipo Tablet**. Estes tópicos são referidos em maior detalhe no subcapítulo 3.12.1 – Publicação Web do Front Panel, sendo a GUI desta versão apresentada na Figura 3.7

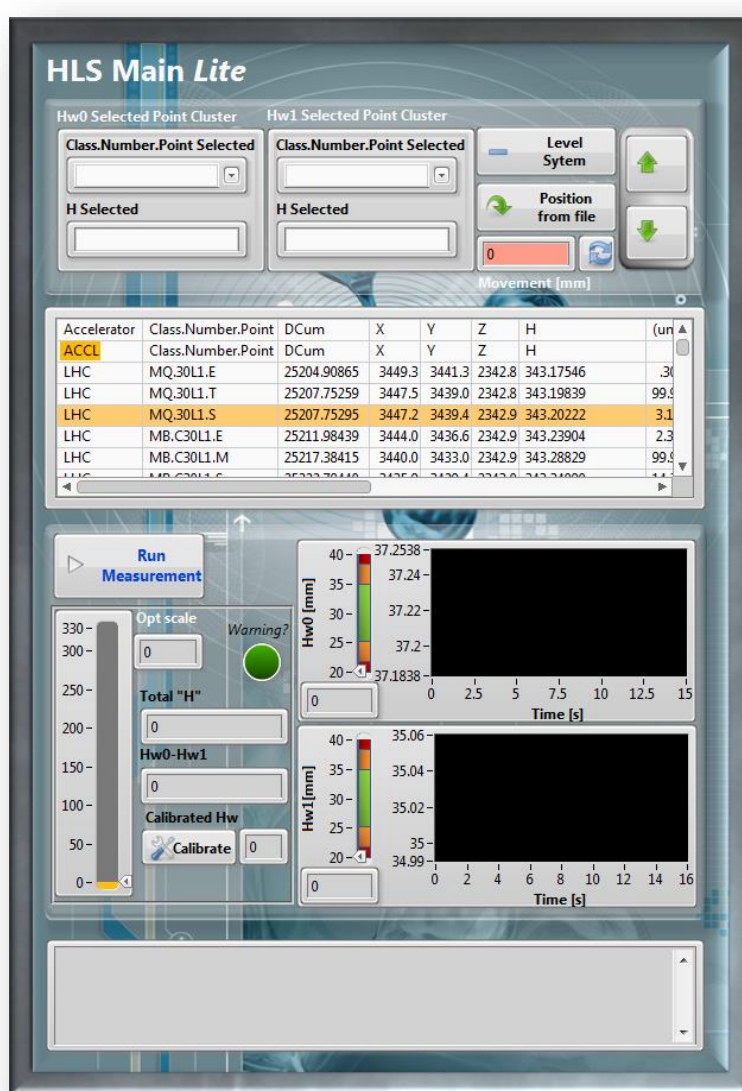


Figura 3.7 – GUI da função *Main_Lite*

Estas versões de controlo remoto consistem em versões minimalistas, mais simples e limpas destinadas a serem apenas uma versão de Operador, focadas apenas na operação do sistema e sem as capacidades de *debugging* e configuração da *Main* completa – uma versão **Lite**, na sua essência mais limitada mas mais intuitiva. Tendo em conta de que certas possibilidades não continuariam ao dispor da aplicação, como, por exemplo, o uso de eventos *pop-up* para o utilizador que são impossíveis de serem reproduzidos numa aplicação de base *web*, uma nova versão foi desenvolvida, tendo a designação de **Main_Lite.vi**. Desta forma, é importante estabelecer uma distinção desta versão, tanto com a versão mais completa **Main_HLS.vi** como com o *conceito* de uma função principal, referido simplesmente como *Main* no decorrer deste relatório. A natureza modular, e de comunicação por mensagens da arquitectura permitiu que ambas as versões utilizem os mesmos módulos, sem a necessidade de adaptar ou formatar os mesmos. Contudo, foi no entanto necessário desenvolver um novo ambiente gráfico e um processo de inicialização que evite a utilização de *pop-ups*, descrito em maior detalhe no subcapítulo seguinte, 3.4.3 – Processos de Inicialização.

3.4.3 – Processos de Inicialização

Os processos de inicialização são controlados pelo **Initializer.vi**, novamente uma Máquina de Estados movida por uma instrução proveniente de uma *Queue*, que contém segmentadas as operações de inicialização individuais. Os *Procedimentos de Inicialização* são contidos numa *lista*, de nome correspondente, que é lida a partir de um VI do tipo de *Functional Global Variable* – FGV denominado **HLS_Macros_FGV**, permitindo que a sequência seja facilmente manipulada, se necessário.

Devido à particularidade de não ser possível a utilização de *pop-ups* para a *Main_Lite*, durante o processo de inicialização é necessário um *identificador* que discirna qual a versão da função “*Main*” que está presentemente a ser executada. Este identificador é obtido através da chamada de **HLS_Is_Lite.vi**, que, lendo o sufixo “*Lite*” do nome da aplicação presentemente a ser executada, determina qual a sequência de inicialização a seguir. Como esperado, a ordem de inicialização dos módulos é inteiramente relevante para a operação do sistema; por exemplo, o módulo *Logger* e os ficheiros *.ini* de configuração dos módulos devem ser executados antes que os módulos em si, que dependem dos mesmos, sejam inicializados. Uma representação dos módulos inicializados pode ser encontrada na Figura 3.5

3.4.3.1 – INI File Manager

Operando como uma FGV, o *INI Manager* é desenhado para procurar e carregar os ficheiros *.ini* utilizados pela aplicação, mantendo-os num registo *Shift-Register*. Cada módulo efectua uma chamada ao *INI Manager* e extrai os dados pertinentes, conforme necessário. Outra solução para estabelecer este mecanismo e distribuir os dados de inicialização de maneira correcta através da aplicação seria ter cada módulo a efectuar um pedido ao *INI Manager* através de uma “*INI Queue*”, recebendo posteriormente a informação através do mesmo canal. Tal introduziria um grau de complexidade adicional na aplicação apenas para depender de uma *Queue* mais, que poderia introduzir instabilidade pela própria natureza passível de ser falível da mesma. Contudo, tal sistema seria mais eficiente, podendo ser capaz de responder a pedidos paralelos se lançado de maneira *não-reentrante*, permitindo várias instâncias simultâneas do *INI Manager*, evitando um ponto de *bottleneck* da aplicação. Uma versão do *INI Manager* que actua desta forma pode ser encontrada nos ficheiros de projecto sob a designação: **INI_Manager_Q_Based.vi**.

O primeiro ficheiro *.ini* a ser lido é o *Folders_Paths.ini*, que contem a estrutura de directórios e localizações de ficheiros da aplicação, incluindo para os demais ficheiros *.ini*. Se inicialmente um caminho para este primeiro ficheiro *.ini* não for definido pelo utilizador, o *INI Manager* irá tentar procurar pelo mesmo, como faria para qualquer outro ficheiro *.ini*. A procura por um ficheiro deste tipo pode ser realizada de duas maneiras: através do Utilizador, onde este manualmente procura e selecciona o ficheiro, ou automaticamente, procurando *recursivamente* dentro do directório da aplicação. Se a pesquisa for inválida, o *INI Manager* irá carregar os valores de defeito para os parâmetros que se encontram definidos programaticamente no código dos próprios Controlos.

Note-se também que o *Folders_Path.ini*, cuja localização pode ser obtida a partir do utilizador, pode conter uma lista de caminhos para os demais ficheiros *.ini* que difere da sua própria. Neste caso, o utilizador *pode*, através do método representado na Figura 3.8, escolher qual a localização a utilizar.

Existe uma distinção entre os processos de inicialização relativa à leitura dos ficheiros *.ini* dependente da versão da função “*Main*” a ser executada. Ao tentar carregar um ficheiro *.ini*, o *INI Manager* irá inquirir através de um *pop-up* de dialogo ao utilizador caso o ficheiro não seja encontrado na localização esperada, ou se mais do que *.ini* do mesmo tipo forem encontrados, à semelhança do apresentado na Figura 3.8. Estas mensagens de texto encontram-se geridas pelo **Dialog_Manager.vi**, que dependendo do vi que lhe chama é capaz de identificar quais as mensagens que deve passar, sendo estas lidas de um ficheiro de texto “*Dialog.txt*” para que possam ser facilmente alteradas.

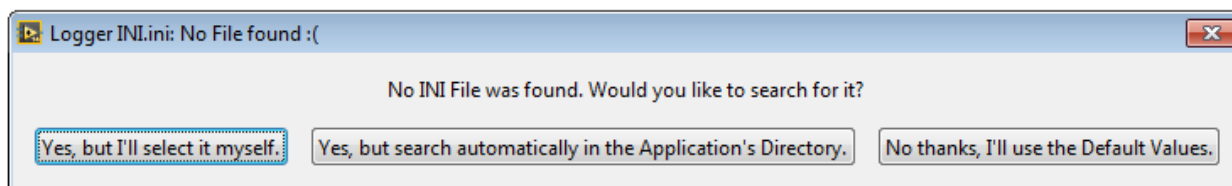


Figura 3.8 – Exemplo de diálogo durante a procura de ficheiros de inicialização

Como estes *pop-ups* de diálogo são interditos em aplicações web, para o caso da *Main_Lite* somos forçados a utilizar uma versão 'Lite' também do próprio *INI Manager*, que segue a mesma lógica de operação mas sem estas funcionalidades interactivas de diálogo.

O procedimento seguido pelo *INI Manager* pode então ser descrito pelo seguinte fluxograma apresentado na Figura 3.9, onde as acções laranja são efectuadas por ambas as versões do *INI Manager*, a vermelho apenas as que são efectuadas pela versão da *Main* completa e a verde as pertencentes ao *INI Manager Lite*.

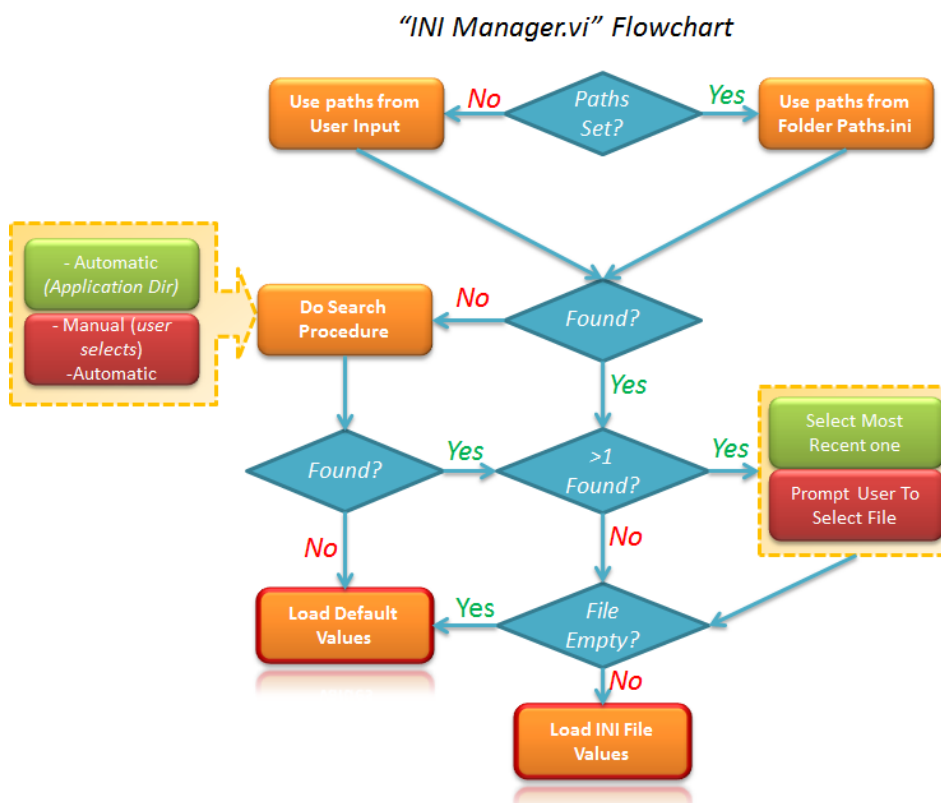


Figura 3.9 – Fluxograma de funcionamento: *INI Manager*

3.4.4 – Eventos dinâmicos com o Módulo: Theo Data Viewer

Outro aspecto que deve ser referido é a interação da função *Main* com o Módulo *Theo Data Viewer*. Por forma fazer o melhor uso deste módulo, que depende pesadamente de eventos de Utilizador para as suas funcionalidades mas de forma remota ao módulo em si, emprega-se a estrutura de eventos existente na própria *Main*. Ao capturar os dados gerados por um dado evento, podemos subsequentemente enviar os mesmos para a estrutura de eventos presente no Módulo *Theo Data Viewer* através de um evento dinâmico e imitar o mesmo como se tivesse sido captado pela mesma. Posteriormente, apenas colectamos a resposta do Módulo ao estímulo e apresentamos os dados na GUI.

Uma representação desta comunicação entre módulos pode ser vista seguidamente na Figura 3.10; aqui podemos observar que ambas as versões **Main_HLS.vi** e **Main_Lite.vi** utilizam o mesmo VI **HLS_Theo_Prep_Event.vi** para enviar os dados do evento e instrução apropriados através da *Theo Queue* para o *Theo Data Viewer*, que processa e envia resposta de volta ao *Display Loop* da função *Main* via a *Display Queue*.

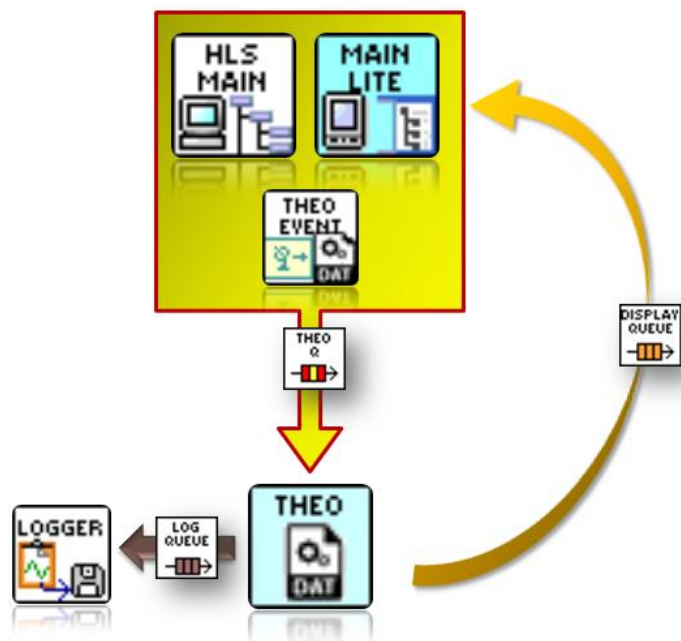


Figura 3.10 – Representação do funcionamento das funções *Main* perante eventos dinâmicos com o módulo *Theo Data Viewer*

3.4.5 – Queues

Sem qualquer utilização de variáveis locais ou globais, o uso de *queues* em conjunto com *FGVs* representam os maiores canais de comunicação utilizados ao longo da aplicação. A referência de cada *Queue* é contida dentro de uma *FGV* individual que mantém a mesma num registo.

Em conjunto com o seu nome, o ícone de uma *Queue* pode ser bastante indicativo quanto a natureza da mesma, respeitando um código de cores que, uma vez familiar age como uma deixa visual útil para a leitura do seguimento da aplicação, conforme apresentado pela Figura 3.11:

- *Queues* ligadas a um Módulo possuem uma cor específica. Apesar de ser meramente indicativo, a cor da *Queue* pode providenciar os indícios da natureza da sua utilização: *Queues* visando o **Hardware** são usualmente **vermelhas** (com a excepção do módulo de controlo *UHLS*, que é **azul**), *Queues* ligadas ao **Display** são **Laranjas**, *Queues* relacionadas com **operações com ficheiros** são **Verdes**, **Dourado** para o **Tratamento de Erros** e as ligadas a serviços auxiliares de suporte à estrutura Cinzento.
- As *Queues* que comunicam directamente a um subcomponente de um dos Módulos principais a partir de outros contem a cor do módulo de destino cercada pela cor do módulo da qual origina a comunicação.

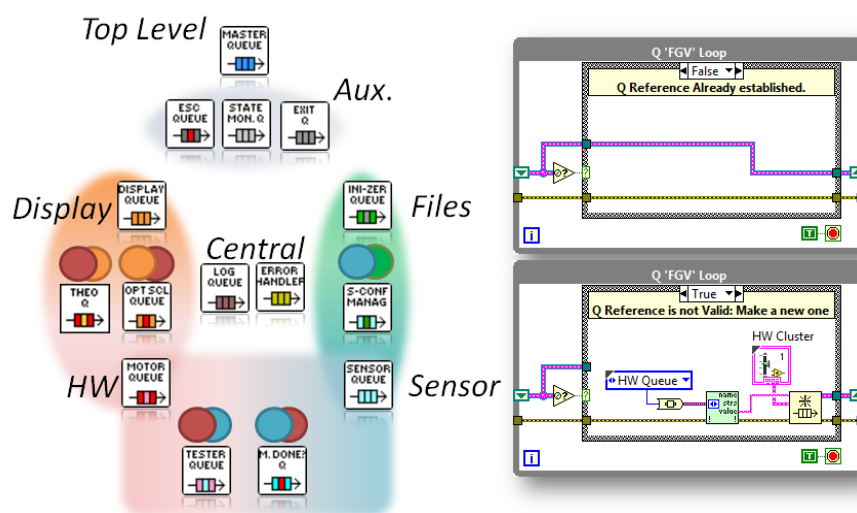


Figura 3.11 – Repartição de Queues por sector (esquerda). Procedimento de alocação de recursos em respeito à referência da Queue (direita)

Sempre que uma chamada é feita a uma FGV de uma *Queue*, esta irá verificar se já existe uma referência aberta e, se sim, utilizar a mesma. Sendo que a função ‘*Obtain Queue*’ irá alocar uma nova *refnum* para uma *Queue* de todas as vezes, mesmo utilizando o nome de uma *Queue* já aberta, este método irá prevenir a alocação deste recurso (que pode até conduzir a um problema de fuga de memória).

Contudo, esta estratégia de optimização pode levar a questões relativas à “atribuição” deste recurso: assim que um VI seja terminado, o LabVIEW irá efectuar uma limpeza de todos os recursos que o VI teria alocado na sua execução e que ainda permaneçam abertos. Isto significa que assim que o módulo que primeiro abriu (“chamou”) a referência para uma dada *Queue* seja terminado, a referência para essa *Queue* torna-se inválida e pode levar ao *Erro 1142 – Invalid Refnum*.

Um **Anchor.vi**, destinado para ler e replicar uma dada *Queue*, destruindo e criando uma nova referência à mesma e assim efectivamente mudando o módulo a qual esta se encontra “ancorada”, foi desenvolvido parcialmente, encontrando-se de momento não acabado.

3.5 – Módulos Top-Level e Auxiliares

Todos os serviços e funcionalidades de suporte dirigidos a infra-estrutura da aplicação em si são inseridos nesta categoria de ditos *Top-Level*, ou Módulo “0”. Os principais constituintes deste módulo são: **Initializer.VI**, anteriormente detalhado, o *Exit Manager* e o *Status Monitor*. Para além destes, outros VIs mais específicos podem ser encontrados aqui, sendo uma breve descrição das suas funcionalidades apresentadas de seguida:



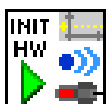
Is_LITE.vi – Identifica a versão da função “Main” através da leitura do sufixo “Lite” do nome da aplicação.



Sort 2D_Array.vi – Utilizado na maioria das *Multicolumn Lisboxes* ao longo da aplicação, irá ordenar um vector de 2D por um dado índice de coluna.



Top-Level_VI_Reference.vi – Uma FGV que contem a referência da Main, tipicamente utilizada de forma independente pelos módulos como forma de verificar o status da Main.



StartUp_Hardware.vi – Chamado pelo **Initializer.VI**, contém os processos de inicialização para os componentes de *hardware* numa simples máquinas de estados.

3.5.1 – *Exit Manager*

Com o aumento da dimensão da aplicação vem um acréscimo na complexidade dos seus subsistemas. Este módulo foi desenhado para conter e processar os processos de saída e terminação dos módulos, liberando a *Main*, que desta forma apenas sinaliza ao *Exit Manager* o comando de saída, com este depois a propagar a acção através de toda a aplicação. Este processo mantém a conformidade com o estilo de *Master(Comando)->Slave(Execução)* da aplicação, “*outsourcing*” o processo a módulos individuais.

Como tal, as principais tarefas do *Exit Manager* consistem em: Executar os procedimentos de terminação, incluindo a limpeza de quaisquer recursos abertos e Verificar o estado da função *Main*: uma paragem anormal levará então a uma saída da aplicação.

Existe um outro módulo que possui autoridade de comunicação para dentro do *Exit Manager*, o *Central Error Handler*, módulo de tratamento de erros, que pode decidir terminar a aplicação como parte do processo de tratamento ou resposta a erros.

Devido a ter vários módulos a correr em paralelo, a ordem de execução para os processos de terminação dos módulos requer tanta atenção quanto os processos de inicialização. Por exemplo, o último modo a ser terminado seria o *Logger*, por forma a registar os processos de saída de todos os outros módulos. O mesmo é verdade para a limpeza dos recursos alocados: conforme detalhado em *Queue References*, terminando um módulo que é o criador original da referência pode levar a um erro em outros módulos que continuem a utilizar a *queue*.

À semelhança do que ocorre com os processos de inicialização, a sequência de operações de saída é determinada desde o **HLS_Macros.vi** que por sua vez a obtém de um ficheiro, por forma a ser manipulada mais facilmente.

Por forma a manter um grau de redundância em caso de falha do próprio *Exit Manager*, cada módulo é capaz de verificar o estado da função *Main*, utilizando a mesma referência e processo que o *Exit Manager* utiliza, tipicamente visto dentro do caso de *timeout* do ciclo do módulo.

3.5.2 – Monitor de Estados - ‘Status Monitor’

Este Módulo auxiliar é lançado por defeito apenas na versão completa da função *Main*, sendo destinado a monitorizar os estados actuais de operação para os diferentes módulos onde se encontra conectado. Tal é conseguido através da conexão dos chamados **Status_Monitor_Minion.vi** ao comando recebido que dita o próximo estado de execução do módulo. Actuando de maneira semelhante a uma “rémora” a este comando, o *Minion* simplesmente “escuta” os pacotes de mensagens que chegam ao módulo, e envia uma cópia do comando que este recebeu para o *Status Monitor*, que posteriormente organiza e mostra a informação ao utilizador, conforme visto pela sua GUI na Figura 3.12.

Sendo que o Status Monitor é mais indicado para operações de *debugging* da aplicação, onde é útil discernir o estado que se encontra um módulo, os *Minions* encontram-se colocados apenas em módulos de *hardware* críticos.



Figura 3.12 – GUI *Status Monitor*

3.6 – Logger: Registo de Dados de Operação

Distribuído pela aplicação, convencionalmente no final da execução de cada iteração do ciclo principal de cada módulo, encontram-se as operações de registo de todos os dados relevantes através da utilização de VIs pertencentes à serie “_toLog.vi”. Estes VIs tipicamente seleccionam e formatam os dados recebidos adequadamente e transmitem os mesmos para o módulo “Logger” em si através da sua *Queue*, conforme visto no princípio de operação representado na Figura 3.13. Com o ficheiro de *log* sendo meramente um simples *.txt* formatado como uma *spreadsheet* e com uma tabulação como separador, a estrutura dos dados guardados segue a expressão de: *Timestamp*, *Source*, *Cmd*, *Message* e *Tag*.

Colocado em execução assincronamente, o *Logger* é uma excepção tanto em relação à sua inicialização como quanto à sua terminação, com cuidados a serem tomados de maneira a que seja **o primeiro a ser inicializado e o último a ser terminado**. Como referido no subcapítulo 3.4.5 – Queues, isto significa que a *Logger Queue* deve estar referenciada ao próprio módulo, ao contrário do que ocorre nos restantes módulos.

Se um comando de Saída é recebido enquanto ainda existirem pacotes de mensagens na *Logger Queue*, o *Logger* irá registar os mesmos – mas apenas estes, antes de sair verdadeiramente da aplicação. Isto significa que quaisquer pacotes que foram colocados na *Queue* depois de este ter recebido o comando de saída serão descartados.

O directório dos registos em si é determinado pelo *File_paths.ini*, como o *Logger* possuindo um limite ao tamanho dos *logs* e número dos mesmos a guardar – depois do limite de tamanho de um log ser excedido, o *Logger* irá criar um novo ficheiro. Por sua vez, se o número de ficheiros for excedido este irá substituir o mais antigo.

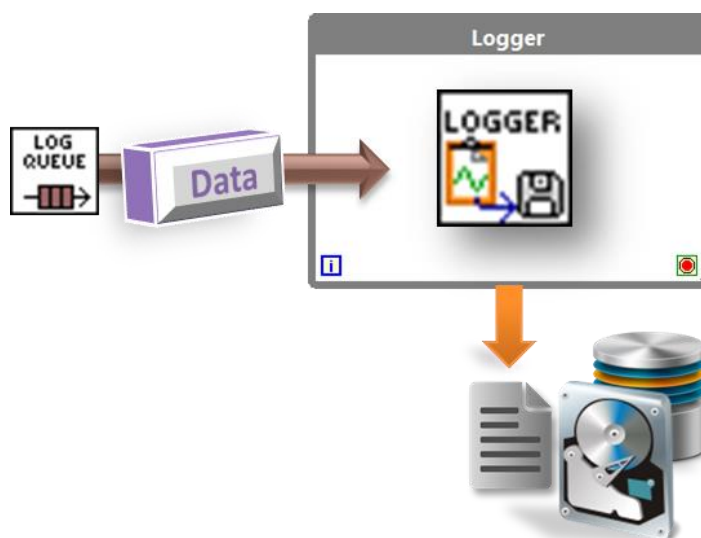


Figura 3.13 – Diagrama de operação do *Logger*

3.6.1 – Visualizador de Logs

Existe a possibilidade de lançar de dentro da aplicação da **Main_HLS.vi**, o **Visualizador de Logs**, cuja GUI se encontra apresentada na Figura 3.14, como uma maneira de apresentar em maior detalhe os conteúdos dos ficheiros *.log*, numa tabela onde o utilizador pode procurar, ordenar ou filtrar por qualquer parâmetro que desejar. De forma permanente, encontra-se também presente na GUI da função *Main* uma consola que apresenta as últimas entradas neste ficheiro sob uma forma melhor descrita como uma lista circular, com as mais antigas a serem substituídas e apresentando as mais recentes primeiro.

Este módulo proporciona então uma maneira mais fácil de procurar entradas específicas, especialmente se conhecido o tipo de entrada que se procura, ou mesmo filtrar por critérios cumulativos as mesmas.

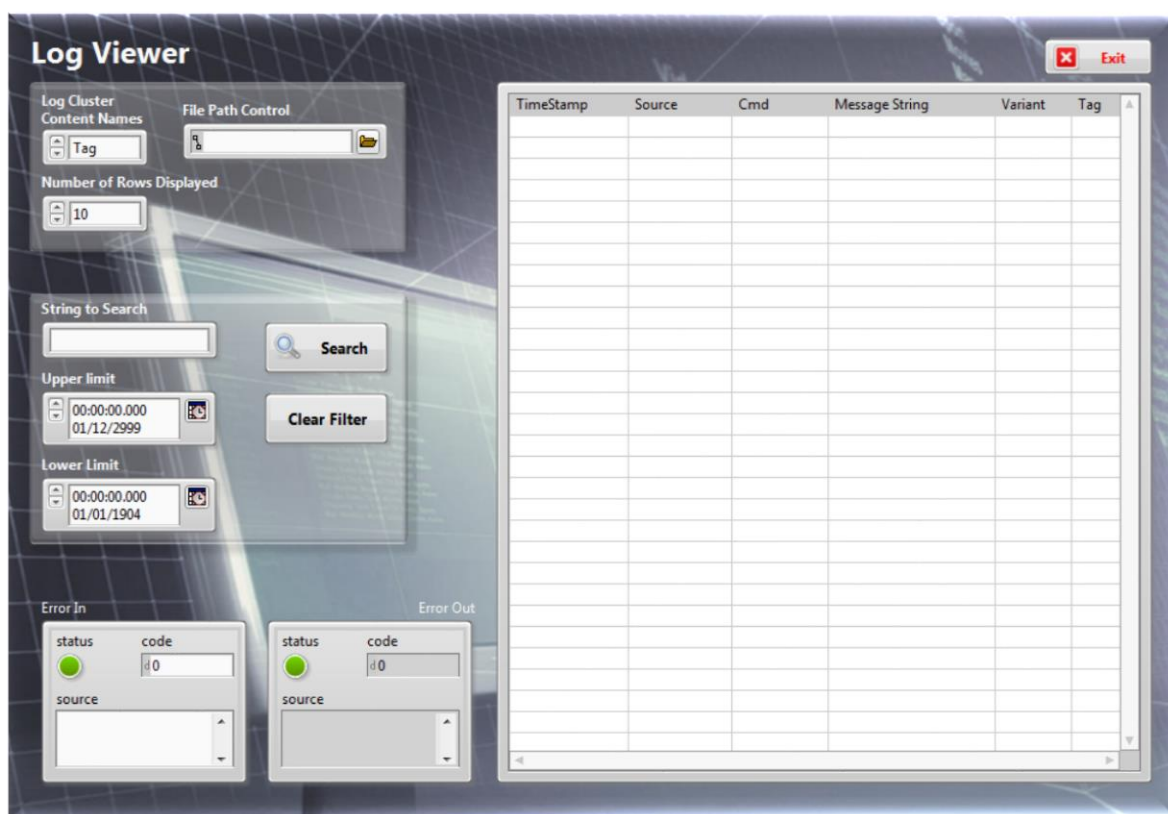


Figura 3.14 – GUI do Visualizador de Logs




3.7 – Mesa de Translação: Motor Zaber Dovetail

O primeiro driver de *hardware* a ser desenvolvido, a Mesa de Translação Zaber Dovetail modelo T-LLS-260mm utiliza comunicação série através de um recurso VISA COM e é conectada ao PXI através de uma conexão RS232. Tratando-se de comunicação série, é passível de ser afectada pelas questões características da mesma, que implica um cuidado especial para garantir que a sessão de comunicação é aberta e fechada da maneira apropriada. Caso tal não aconteça, corremos o risco de deixar uma sessão aberta (“não-terminada”) que bloqueie novas ligações de serem estabelecidas.

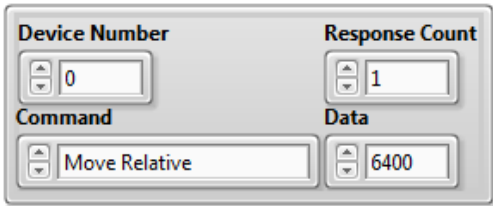
Todos os detalhes acerca do protocolo de comunicação, bem como um guia de implementação e dos comandos utilizáveis encontra-se disponível em:

<http://www.zaber.com/wiki/Manuals/T-LLS>

O fabricante providenciou também uma serie de VIs de controlo oficiais através da biblioteca Zaber Z Series que pode ser obtida internamente pelo LabVIEW em *LabVIEW's Help-> Find Instrument Driver*. Esta *palette* permite então a utilização dos seguintes Vis:

-   **Initialize e Close Com** – utilizados para *Iniciar* e *Terminar* a sessão de comunicação com o equipamento;
-  **Request Command** – utilizado para enviar comandos para o equipamento. O comando enviado precisa de conter o seguinte formato, sendo um *cluster* com os seguintes campos:

- **Device Number**
- **Response Count**
- **Command**
- **Data**



Device Number	Response Count
0	1
Command	Data
Move Relative	6400

Enquanto que os demais campos são auto evidentes, note-se que o campo “*Data*”, referente à informação necessária de um dado comando, quando referente a posições de movimentos, representa o número de *micro-passos* do motor. A razão de conversão é, por defeito, 6400 para 1mm. Uma exposição mais alargada dos comandos pode encontrada na página do fabricante.

3.7.1 – HLS Motor Main

O conceito deste módulo segue os princípios já mencionados de paralelismo, sendo lançado assincronamente, e funcionando como uma *máquina de estados* movida por uma *Queue*. A sua principal responsabilidade é lidar com as operações de movimento do sistema, através do controlo do Zaber.

Existem duas formas principais de movimento: Manual ou Automático. Movimento Manual é feito através de uma mecânica de manter o botão da direcção desejada pressionado, conforme visto na descrição da GUI da Main, enquanto que o movimento automático refere-se a uma sequência de movimentos definidos programaticamente.

O operador pode também seleccionar um comando próprio do Zaber da sua lista de comandos acessível e enviar o mesmo directamente para o equipamento, sendo que deve estar acompanhado dos parâmetros adequados. Desta forma é possível alterar parâmetros como a *Velocidade*, *Limite máximo de Posição*, ou *Tamanho do Passo*. Contudo, nenhuma necessidade de alterar estes parâmetros está prevista, sendo que apenas deverá ser efectuada consultando a documentação apropriada. Caso seja utilizado o Comando de *Restaurar as Definições*, é de notar que os valores configurados para dentro da memória interna do equipamento, como o limite máximo de operação de segurança (que estabelece os limites de deslocamento para que não choque com os limites da estrutura) devem ser repostos em separado.

3.7.1.1 - Put in Place: Position from File

Esta sequência é a primeira parte do processo de colocar automaticamente os sensores dentro do seu alcance de operação. Ao longo deste documento, será seguida a designação de cápsula Estática e Móvel, referidas como Hw0 e Hw1 respectivamente para que corresponda a leitura de dados proveniente do sensor.

O ficheiro *Theo.Dat* contém aproximadamente a posição das diferenças de altura entre os Ímanes, proveniente de campanhas de medição anteriores que estão bem dentro da precisão na ordem de milímetros. Após definir um parâmetro de calibração para o *offset* de posição entre as duas cápsulas, com o valor esperado de diferença entre os pontos lido a partir do ficheiro *Theo.dat*, é possível posicionar as duas para que estejam dentro do seu alcance operacional, que foi medido experimentalmente em 4cm. Por exemplo, se a diferença entre os dois pontos for lida como de 15cm, apenas temos de posicionar a cápsula móvel para compensar esta distância.

Note-se que apenas uma das cápsulas é móvel, e que para o alinhamento ser possível a cápsula estática não deve estar posicionada mais abaixo do que o mínimo da cápsula móvel. Esta questão é abordada em maior detalhe no subcapítulo 3.11 – Theo Data Viewer.

O processo pode então ser descrito nos seguintes passos: após receber o Comando e os dados de posição já validados, o sistema verifica a posição da cápsula móvel através da Régua Óptica. Se este não se encontra dentro de um limite de proximidade, então a cápsula móvel deve mover-se continuamente na direcção correcta para compensar a diferença de altura entre as duas. Continuando a monitorizar a sua posição, quando o destino final for alcançado ou quando alguma das protecções de segurança forem activadas (Destino ultrapassado ou Erro na leitura da Régua Óptica) o sistema parará, e caso o movimento tenha sido bem-sucedido, enviará um sinal de comando para uma Electroválvula (EV) que regula o canal de comunicação do líquido, abrindo a mesma. A descrição deste processo encontra-se também sob a forma do fluxograma presente na Figura 3.15

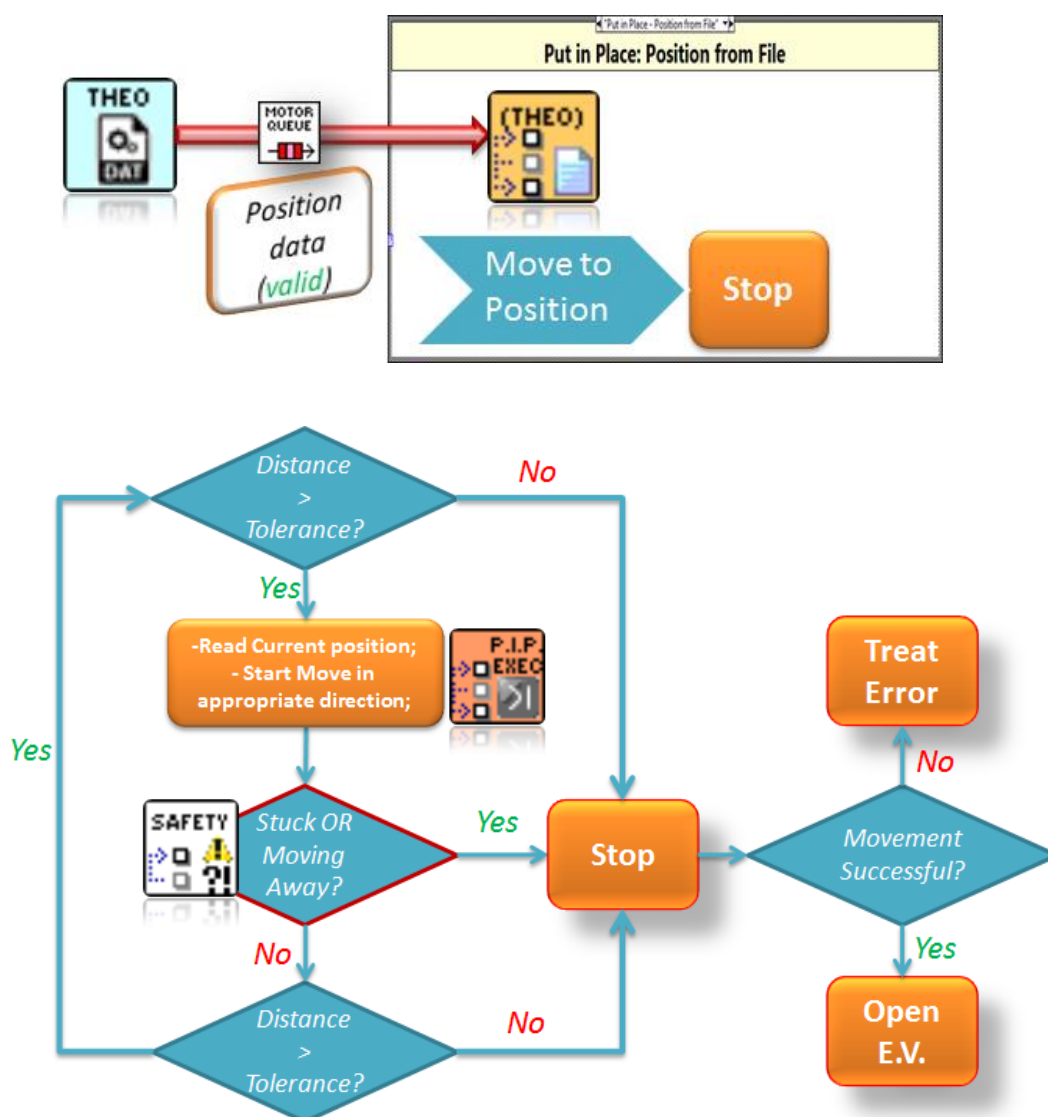


Figura 3.15 – Princípio de Operação e Fluxograma de Funcionamento da função:
Put in Place: Position from File

O Controlo da EV é realizado através de uma placa de *Data Aquisition* - DAQ multifunções do tipo USB-6009, de 12 portas Digitais I/O, e utilizando sub-VIs de controlo da palete com o mesmo nome, DAQmx. Para o accionamento da EV é necessário conectar um sinal digital, que deve ser mantido a “1” para que esta continue aberta.

Inicialmente, a EV utilizada necessitava de uma pressão do líquido demasiado elevada para permitir a comunicação do mesmo sob as situações de utilização deste sistema, necessária para abrir a membrana reguladora. Este factor, conjugado com uma abertura demasiado pequena, que levaria a um tempo de estabilização acrescido, levou à substituição da mesma por outro modelo, apresentado na Figura 3.16. Não obstante, deve ser notado que o tempo de estabilização pode ser acrescido devido às características incorrectas da EV.



Figura 3.16 – NI DAQ USB 6009 e Electroválvula

3.7.1.2 - Put in Place: Fine

Uma vez dentro do alcance de operação, a EV é aberta permitindo que o líquido flua livremente entre as duas cápsulas (ou recipientes), e possibilitando uma medição. Contudo, para garantir a máxima fidelidade e precisão dos resultados, os níveis de água de cada cápsula devem estar o mais aproximado possível, *i.e.* as cápsulas devem estar niveladas.

Para tal, em concordância com a representação da Figura 3.17, primeiro é efectuada uma medição do sensor em ordem a determinar o nível de água em cada cápsula (H_w), e determinado a diferença entre estes. Este valor é a diferença de altura entre as cápsulas que deve ser compensada pela cápsula Móvel.

A partir deste ponto, o processo progride da mesma forma que o anteriormente visto para o outro movimento *Put in Place – Position from Theo.*

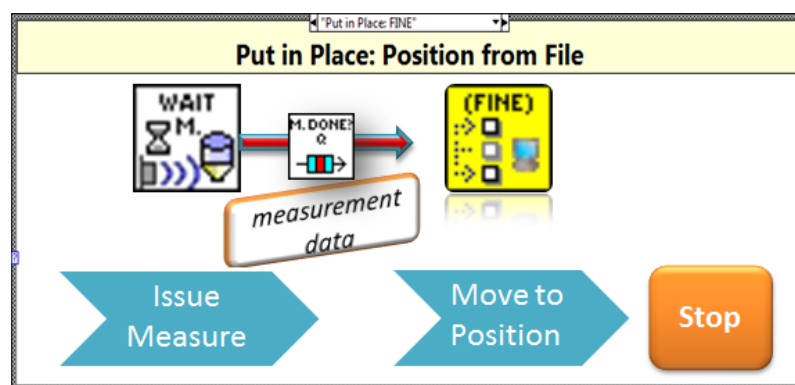


Figura 3.17 – Diagrama de Operação da função **Put in Place: Fine**

3.7.1.3 – Movimento de Calibração

Devido à própria natureza da tecnologia do sistema de medição, que utiliza um líquido entre dois vasos comunicantes, existe a possibilidade de ocorrerem erros operacionais que podem comprometer a validade de toda a medição; por exemplo, casos os vasos não estejam a comunicar devido à presença de bolhas de ar no tubo que o bloqueiem, ou se o líquido ainda se encontrar a estabilizar.

Embora o tratamento de medições inválidas seja abordado no subcapítulo referente 3.9.1 – Tratamento Estatístico, existiram várias instâncias documentadas onde o comportamento do sistema de medição produziu uma série de valores aceitáveis e credíveis, muitas vezes centrados a volta dos 36mm, mas que na verdade são falsos. Derivados do comportamento das reflexões dentro da cápsula assimétrica, com uma alta correlação ao alinhamento da peça de referência, as causas deste fenómeno não são totalmente previsíveis. Desta forma, um teste de calibração foi estabelecido como outra ferramenta de discernir a validade das medições. Primeiramente, é efectuada uma medição, e se validada pelo tratamento usual, o

teste prossegue. Seguidamente, um movimento ondular é efectuado, acabando 2mm acima da posição inicial, seguido por uma segunda medição. Podemos assim, através da análise dos dados da Régua Óptica dos dois momentos, precisar com elevada certeza a distância do movimento efectuado.

Se os sensores não corroborarem esta diferença de altura, Ou se a soma total dos níveis de água das duas cápsulas para as duas medições não for constante, Ou se a correlação do movimento ondulante não for detectada (indicando a comunicação do liquido), tudo dentro de uma dada tolerância, o Diagnóstico de Calibração é inválido e o utilizador notificado. Se, contudo, este não e o caso e a série de medições é considerada válida, então o Nível de Água Total, soma das duas cápsulas, será registado. Sendo que este valor manter-se-á constante ao longo das medições, com apenas os valores individuais de cada cápsula a variarem, este constitui uma referência indicativa da validade de uma medição. Esta descrição do processo pode ser expressa com o fluxograma seguinte constituinte da Figura 3.18.

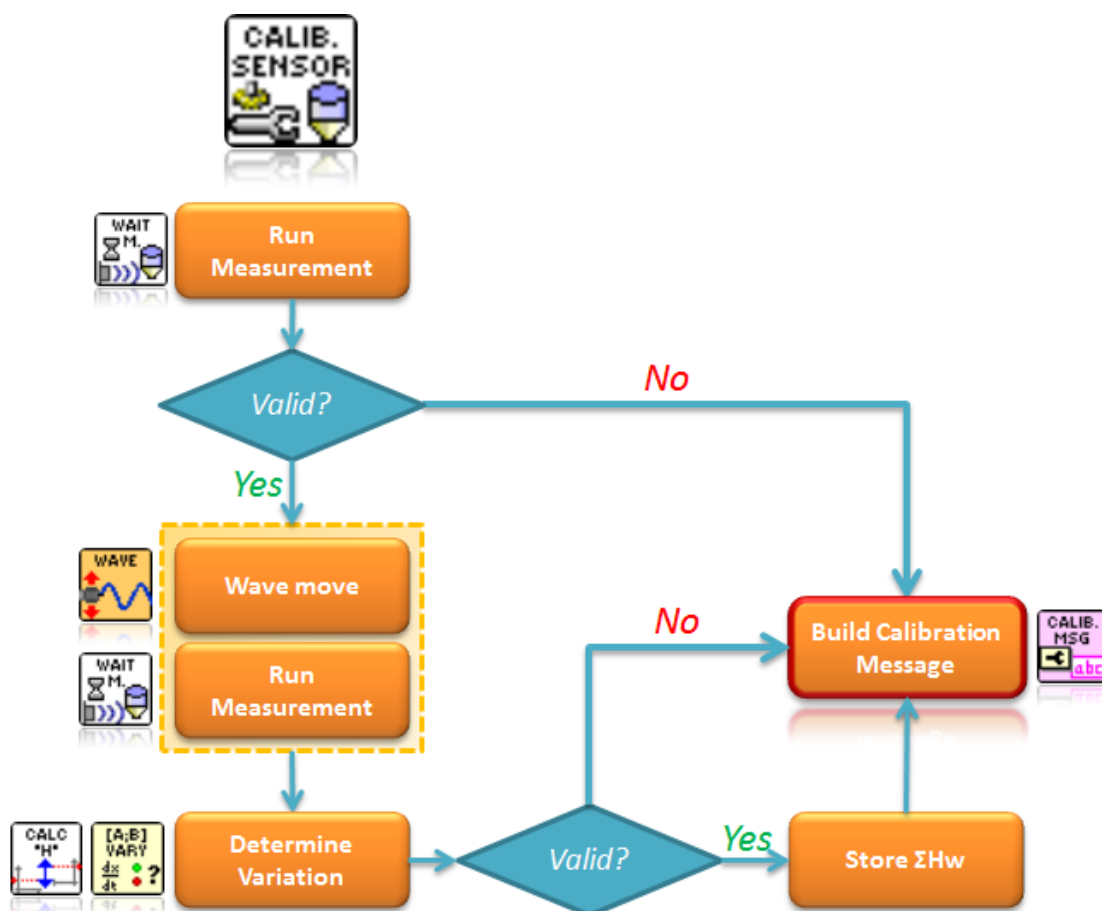


Figura 3.18 – Fluxograma de Funcionamento: **Movimento de Calibração**

3.8 – Régua Escala Óptica

O sensor óptico de alta precisão do tipo Régua Linear, que lê os dados de posição da cápsula móvel disponibilizando-os ao restante do sistema, é desenhado de forma diferente aos outros módulos. Sendo um aspecto crítico da operação de todo o sistema, a leitura da posição da cápsula móvel apenas pode ser realizada através de monitorização (*polling*) constante. O comportamento do módulo é assim, bastante simples: Inicializar e Continuar a ler os dados relativos à posição até à ordem de paragem. O procedimento de inicialização é colocado dentro de uma simples máquina de estados, com os canais de comunicação do módulo a serem simplesmente a *Queue Optical Scale*, relativamente a mensagens de entrada, e a *Queue* de Display, para as mensagens de saída, conforme apresentado na Figura 3.19, contendo os dados de posição lidos.

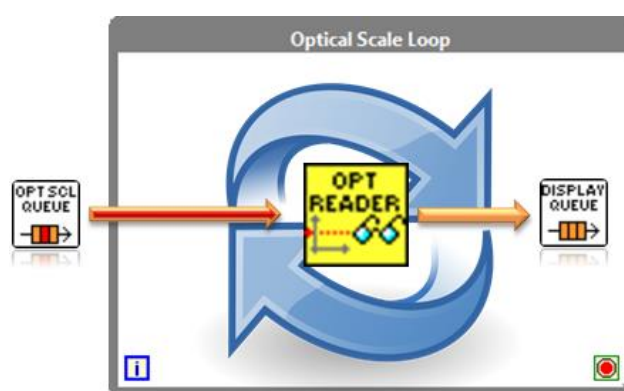


Figura 3.19 – Representação de Operação do Módulo de controlo da Régua Óptica

A complexidade presente no módulo deve-se ao desenvolvimento dos *drivers* para LabVIEW. Em primeiro lugar, para utilizar o adaptador de BiSS para USB é necessário a presença de um controlador específico, *USB_MB3U*, instalado no sistema operativo base (neste caso, o PXI OS: Windows 7). Este é facultado através do *website* do fabricante e também encontrado no directório do projecto, em Documentation\Files.

O outro requisito é a biblioteca *.lvlib BiSS1SL_Interface.dll* (5.2), como uma forma de aceder às funções contidas no ficheiro *.dll* de mesmo nome. É através desta *.dll* que a comunicação e controlo são estabelecidos com o dispositivo.

Os detalhes para as principais funções e parâmetros foram determinados tomando como base a sua documentação própria da BiSS.dll e encontram-se expostos na Tabela 3.3:

Tabela 3.3 – Parâmetros BiSS utilizados

Parâmetro	VI	Valor	Descrição
ulInterface	Set Interface	4	USB Port
eBiSS_PROTOCOL	Set_COM_Parameters	0	BiSS C transmission
eBiSS_CLKENI	Set_COM_Parameters	0	Clock: '0' = External; '1' = internal
eBiSS_FREQ_SCD	Set_COM_Parameters	4	Sensor Data Frequency
eBiSS_FREQ_REG	Set_COM_Parameters	2	Register Data frequency
eBiSS_FREQ_AGS	Set_COM_Parameters	63	AutoGetSens Frequency
eBiSS_SCD_LENGHT	Set_FRAME_Parameters	34	Data range of the Single Cycle Data CRC-bits
eBiSS_SCD_CRC_POLY	Set_FRAME_Parameters	67	Single Cycle Data CRC-polynomial
eBiSS_SCD_CRC_INVERT	Set_FRAME_Parameters	1	'0' = not inverted; '1' = inverted (CRC-bits)
eBiSS_SCD_CRC_CONTENT_COUNT	Set_FRAME_Parameters	63	Number of SCD contents

No caso da sessão de comunicação série com o dispositivo não ser fechada, *i.e.* **BiSS_Close.vi** não ser devidamente executado, durante a próxima vez em que a aplicação for iniciada, então a função **BiSS_Open.vi** irá devolver um parâmetro “*handle*” (que representa o ID único do dispositivo) incrementado por “1”, em relação ao valor anterior. O facto de este parâmetro não ser “0” irá resultar num *Erro de Código* na ordem dos “600x”, que representa a gama reservada para Escala Óptica. Como resolução a este problema, a ligação USB deve ser desconectada e conectada manualmente, existindo também registos de situações onde um *reboot* para o LabVIEW foi necessário (versão 2013). Um sub-VI foi desenvolvido por forma a imitar esta operação, desactivando e voltando a activar todas as conexões USB, depois das quais uma inicialização da Escala Óptica seria o bastante para corrigir o comportamento.

Este VI é actualmente chamado de dentro de um VI reparador - “*Fixer VI*”, conforme será abordado em maior detalhe no subcapítulo 3.10 – Tratamento de Erros.

3.9 – Sensor de Nivelamento Hidrostático Ultra-sônico (UHLS)

O mais complexo e extenso dos Módulos que constituem a aplicação, este subsistema controla o protótipo do Sensor Ultra-sônico de Nivelamento e as diversas *RecBoards* que nele residem. Ao contrário dos outros módulos, este foi desenvolvido de forma a poder ser inicializado singularmente fora da *Main*, tendo o seu próprio painel de instrumentos e interface gráfica conforme apresentado na Figura 3.20. Este painel permite ao Operador aceder a opções adicionais mais técnicas do módulo, como é o caso para o seu Gestor de Configurações - *Configuration Manager*.

Distinto dos outros módulos de *hardware*, o seu protocolo de comunicação é através de TCP/IP, sendo que a o endereço de IP do PXI deve ser colocado dentro da mesma sub-rede que o MPE.

As informações relativas aos telegramas de comando e o contudo de suas mensagens pode ser encontrada no manual próprio providenciado pelo fabricante DESY [10].

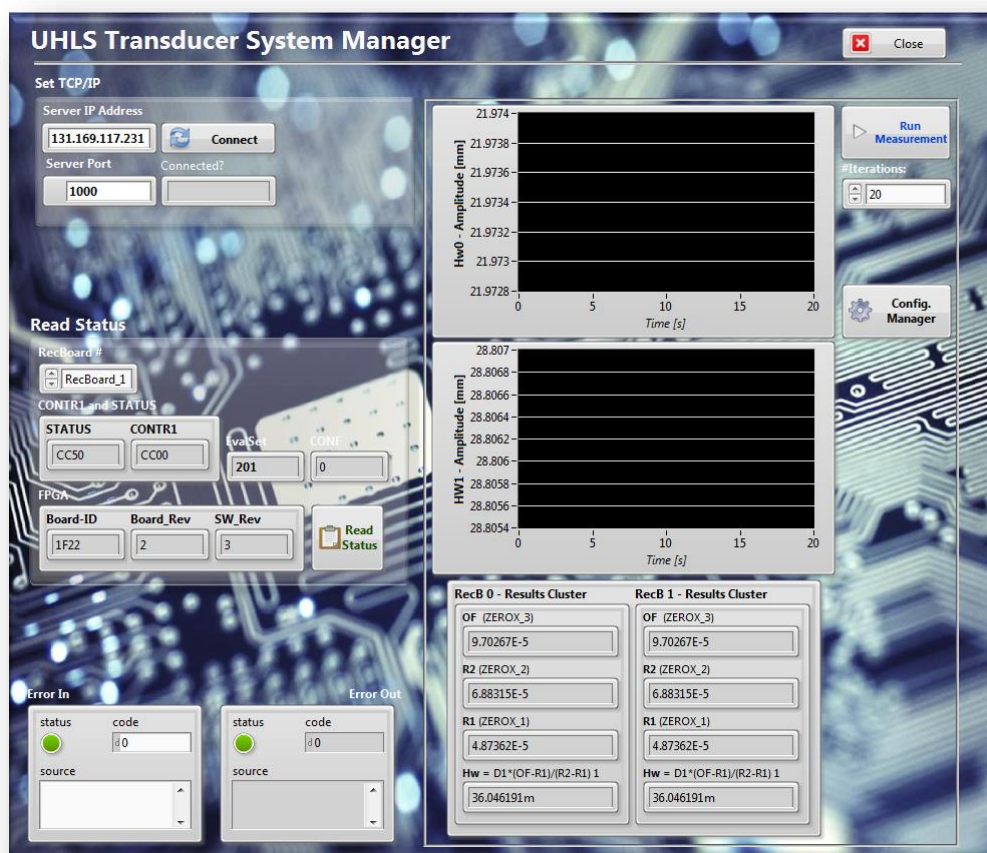


Figura 3.20 – GUI do Módulo *UHLS*, apresentando as ‘Opções Avançadas’

A principal operação realizada consiste em efectuar uma medição do nível de água presente nas duas cápsulas pelos sensores. Denominada de “*Run Measurement*”, esta é construída através do seguinte processo: primeiro é enviado o telegrama apropriado de comando, com os parâmetros respectivos, seguido pela leitura dos dados e finalmente o seu tratamento. O processo e a sua sequência podem assim ser visualizados no seguinte diagrama da Figura 3.21, indicando os principais sub-VIs:

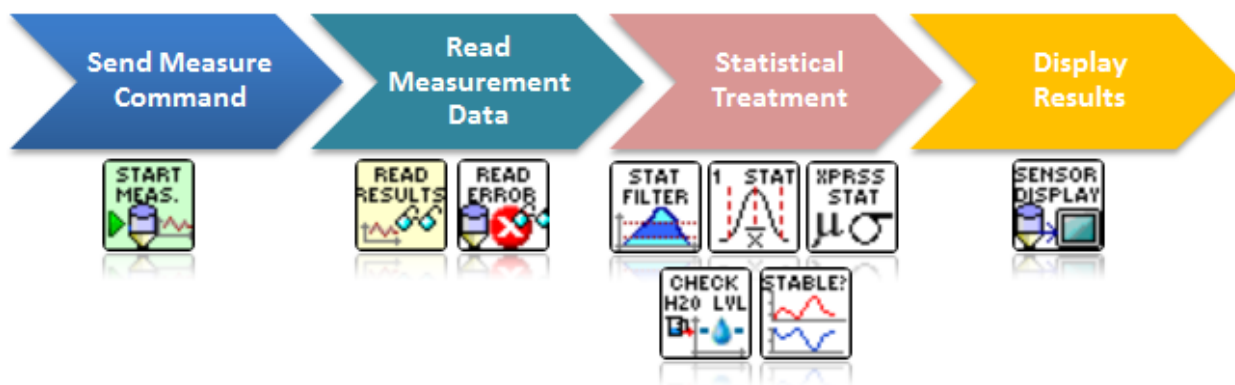


Figura 3.21 – Sequência de Operação do módulo UHLS ao efectuar uma medição

3.9.1 – Tratamento Estatístico

Em medições com precisões tão elevadas existe um grande número de factores a considerar que podem afectar a validade da medição. Isto é tanto mais verdade num sistema com as características físicas do HLS, nomeadamente em termos da estabilidade do líquido ou da assimetria da cápsula em si. O maior factor que pode afectar a validade de uma medição é o **alinhamento da peça de referência**, algo demonstrado sucessivamente durante testes práticos ao longo do desenvolvimento do projecto, o alinhamento e orientação da peça de referência é crítico para o correcto funcionamento do sensor. Além de ser caracteristicamente intrínseco da natureza do processo em si, factores como erro na medição dos parâmetros da peça de referência, imperfeições ou fissuras na mesma, bem como a assimetria da própria cápsula sejam factores que justificam a importância do alinhamento da peça de referência.

O uso da técnica Impulso-Eco ultra-sónica para determinar a altura do líquido na cápsula é também sensível à estabilização do líquido, além de que, conforme mencionado para o processo de calibração referido em 3.7.1.3 – Movimento de Calibração, existe também o risco de valores falsos credíveis serem lidos.

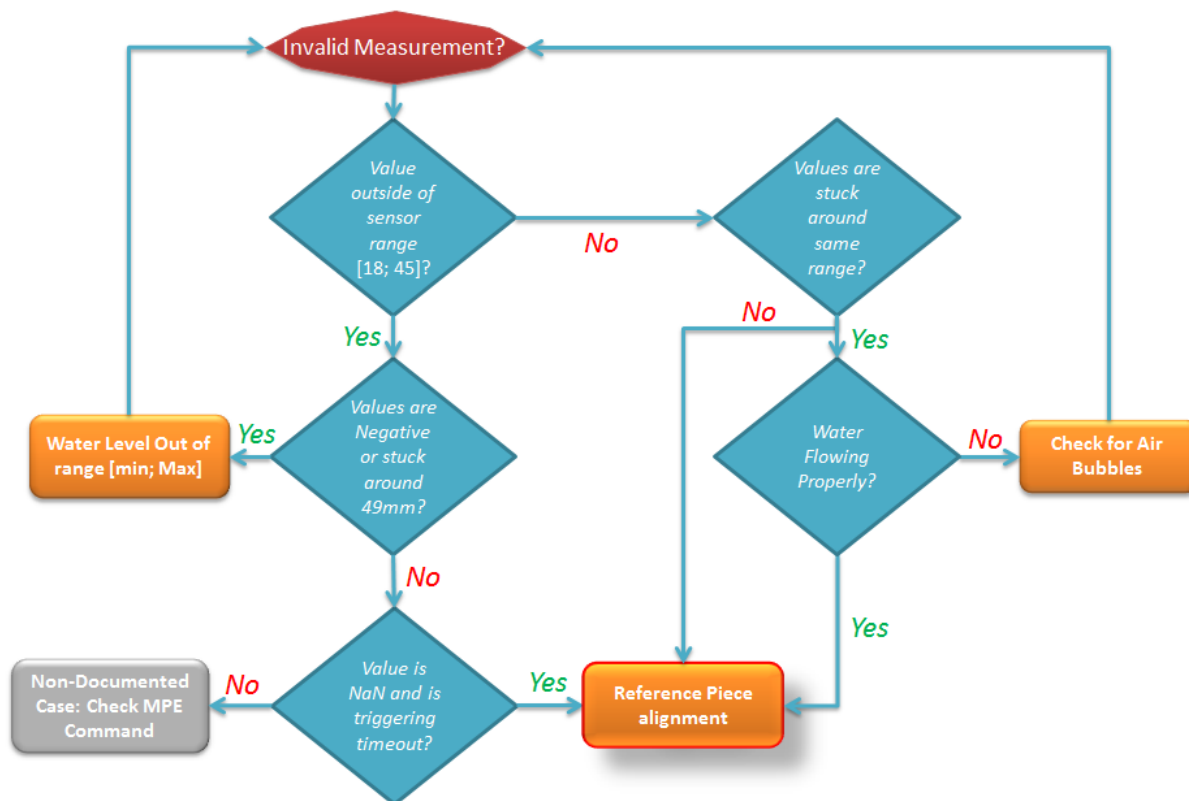


Figura 3.22 – Fluxograma com os procedimentos de Resolução de Erros

Devido a todos estes factores que podem afectar a precisão e validade da medição, o **tratamento estatístico dos dados** é concretizado com base numa estratégia de acção em múltiplos níveis. A primeira etapa consiste em tomar uma medição não como um ponto de medida isolado, mas uma *Série* de medições, com um intervalo entre medições para estabelecer uma média entre estes. Neste ponto, é pertinente notar que pode ocorrer uma situação em que um sensor devolva um valor infinito (sem reflexão) e que resulta num *timeout* no subsistema de leitura de dados, levando à introdução de uma ligeira latência no processo.

Seguidamente, em concordância com os procedimentos habituais de medições empregues pelo *Survey Group*, são retirados da série os primeiros e os últimos 10% dos valores.

A Figura 3.22 representa um fluxograma de resolução de problemas a ser seguido pelo Operador que procura remover o factor da experiência do mesmo com o equipamento a ser determinante na resolução de problemas que possam surgir.

A próxima etapa consiste em verificar a Média e Desvio Padrão da série. Se o desvio Padrão exceder um limite estabelecido, então procedemos a verificar a diferença de cada ponto à Média: caso um valor individual exceda um valor máximo, este é então removido. Se mais do que uma determinada percentagem for removida de uma Série, então a mesma é declarada inválida. Caso contrário, a série reconstruída é apresentada através de um gráfico em conjunto com os seus parâmetros de Média e Desvio Padrão para o nível de água Hw.

Seguidamente, através da análise da correlação entre as duas Séries de Medições, (uma para cada cápsula), podemos determinar se o nível do líquido encontra-se a estabilizar: se tal for verdade, observa-se que as duas medições possuem uma correlação inversa entre elas – uma subida no nível de água de uma cápsula irá significar uma descida proporcional do mesmo na outra. Através de processos experimentais, verifica-se que o estabelecimento de um limite de correlação na ordem dos $|0,6|$ representa um comportamento adequado.

Seguidamente, na Figura 3.23 é exposto o fluxograma representativo de todo o processo de tratamento estatístico anteriormente descrito. Desta forma, em conjunto com os métodos de diagnóstico, obtemos um maior grau de automatização e autocorreção das medições que permite isentar ao máximo o Operador de possuir uma experiência de operação especializada com o sistema que lhe permita inferir sobre a validade das mesmas.

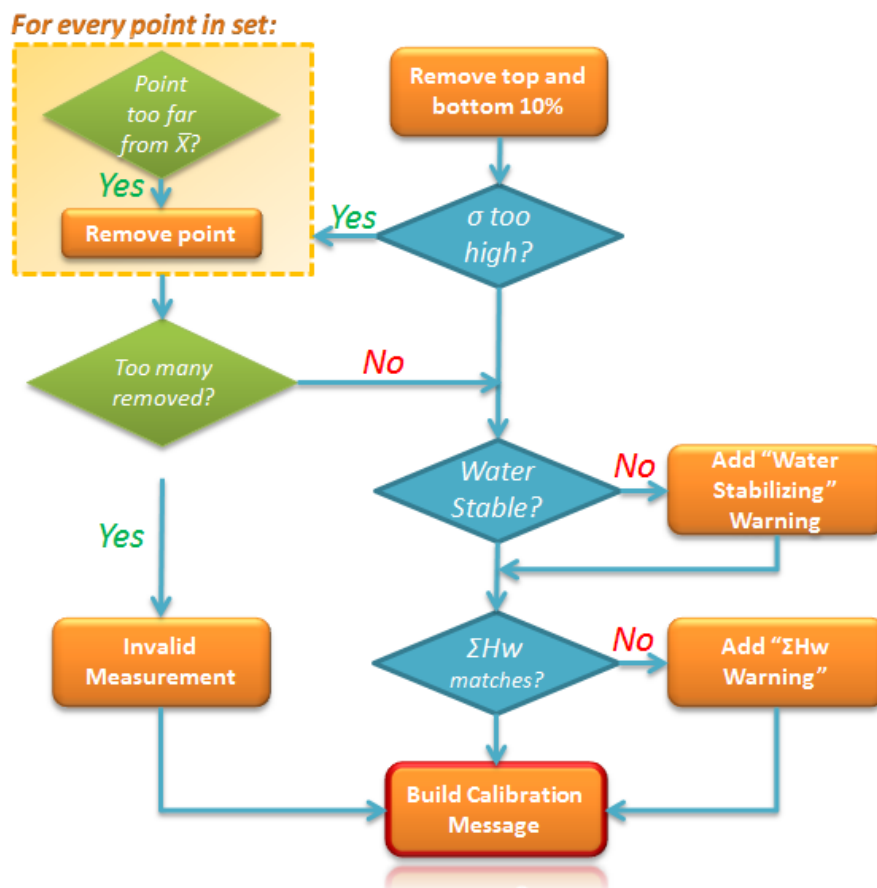


Figura 3.23 - Fluxograma dos procedimentos de Tratamento Estatístico das medições

3.9.2 – Gestor de Configuração do UHLS (Configuration Manager)

Acessível apenas através da GUI do Módulo UHLS, o Gestor de Configuração do UHLS – *Configuration Manager* permite manipular todos os parâmetros de configuração para o MPE, desde a amplitude do impulso até aos limites das janelas de observação das reflexões, conforme pode ser observado na sua GUI representada na Figura 3.24.

A descrição destes parâmetros encontra-se explicada em maior detalhe na documentação pertinente em anexo, e embora sejam possíveis de ser alterados a fim de corrigir uma dada situação ou aumentar a precisão, qualquer alteração no estado do sistema, incluindo movimento, iria implicar um novo ajuste dos parâmetros. Dado a natureza mais complexa e não linear dos mesmos, é recomendado que sejam mantidos nos valores de *default*, já expressando a melhor configuração possível. Se um problema com a medição continuar a persistir, é muito provável que o processo seja corrigível com o alinhamento da peça de referência, conforme visto no fluxograma de resolução de problemas para a medição na Figura 3.22

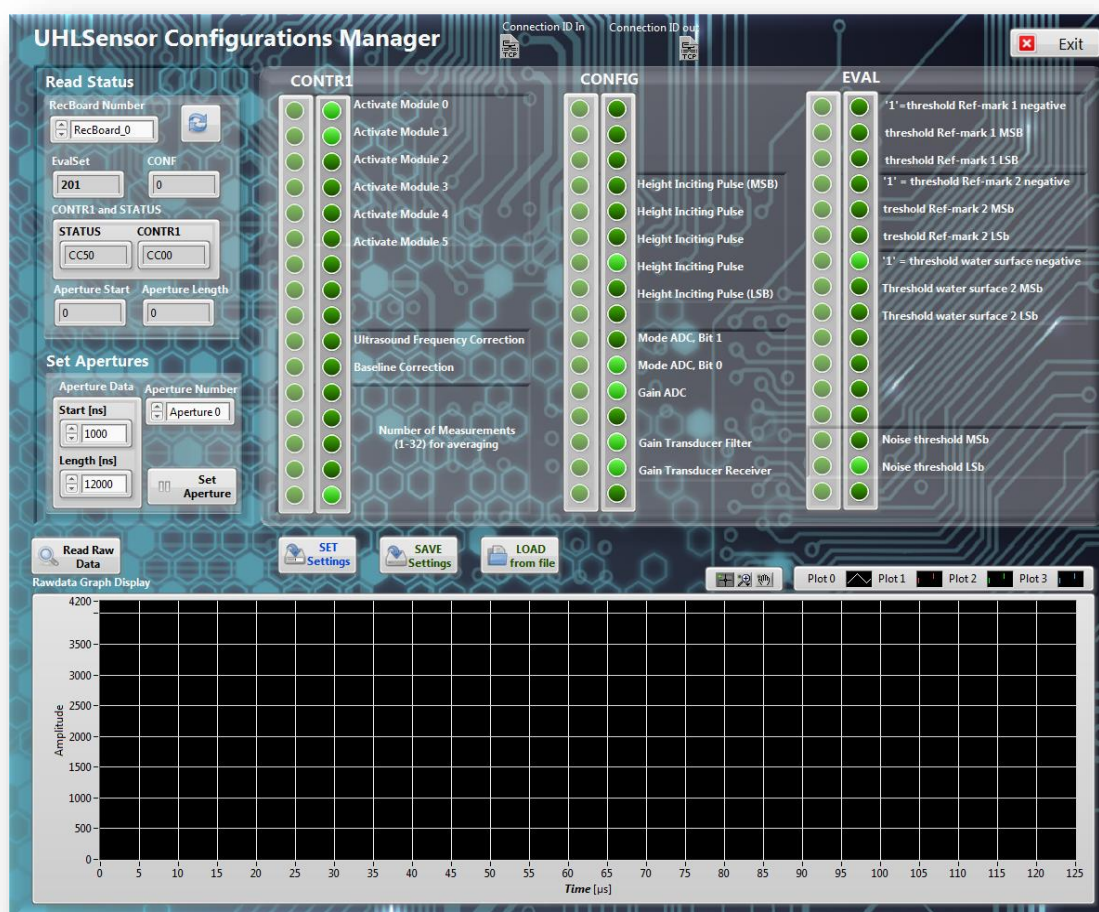


Figura 3.24 – GUI do Gestor de Configurações do Módulo UHLS

O *Configuration Manager* pode também ser utilizado como uma poderosa ferramenta de diagnóstico através da função *Read Raw Data*, permitindo a leitura dos dados directamente da medição. Esta visualização, exemplificada na Figura 3.25, pode ser utilizada como forma intuitiva de *debug* do sistema, onde os erros ou falhas nas reflexões podem ser facilmente visualizados. O Operador pode então realizar uma correcção à peça de alinhamento e observar de que forma são afectados os resultados.

Conforme a documentação, existem certas condições que definem a qualidade da medição que podem ser facilmente identificadas neste modo de visualização:

- **Alinhamento Mecânico:** Idealmente, todas as três reflexões, R1, R2 e OF devem ter a mesma amplitude. Devido a mal alinhamentos impossíveis de serem eliminados, ocorrem ecos indesejados da última reflexão OF, que devem possuir valores contidos abaixo dos da Reflexão mais pequena. Para que não ocorra nenhuma interpretação incorrecta dos resultados que comprometa a precisão, a reflexão mais pequena deve ser maior do que 20% da maior reflexão.
- **Ganho:** Todas as 3 reflexões do sinal devem estar dentro do alcance dinâmico do ADC de 12 bits, seja, maiores que 0 e menores que 4095. Se tal não for o caso, o ganho deve ser reduzido. A menor amplitude deve estar acima de 50 incrementos do ADC, (2100 na figura), caso contrário o ganho deve ser aumentado.
- **O ponto de “Zero Crossing”** (cruzamento do 0 sobre o eixo Oy) deve ser definido por forma a ser utilizado pelo microcontrolador como o Tempo de Propagação do respectivo sinal. Deve ser definido assim um limite em percentagem da amplitude total, e que pode ser de sinal positivo ou negativo. O Microcontrolador irá calcular o ponto de cruzamento com base neste sinal (transição positivo ou negativo).

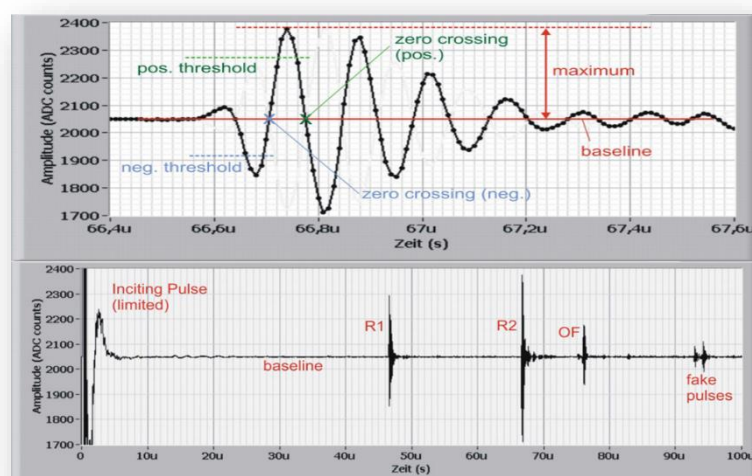


Figura 3.25 – Exemplo da leitura de dados através da funcionalidade *Read Raw*

Existem outros fenómenos documentados que, embora em menor escala, podem influenciar a operação do sistema. Sob períodos de tempo mais prolongados, verifica-se uma perda no volume de água total do sistema, que, se demasiado pronunciada, pode levar a que as cápsulas sejam colocadas fora do alcance de operação do sensor. Apesar de este fenómeno não ser relevante em termos de medições operacionais, ocorrendo ao longo de janelas temporais bem mais longas, não obstante foi estudado e documentado em condições controladas, com os resultados expostos na Figura 3.26.

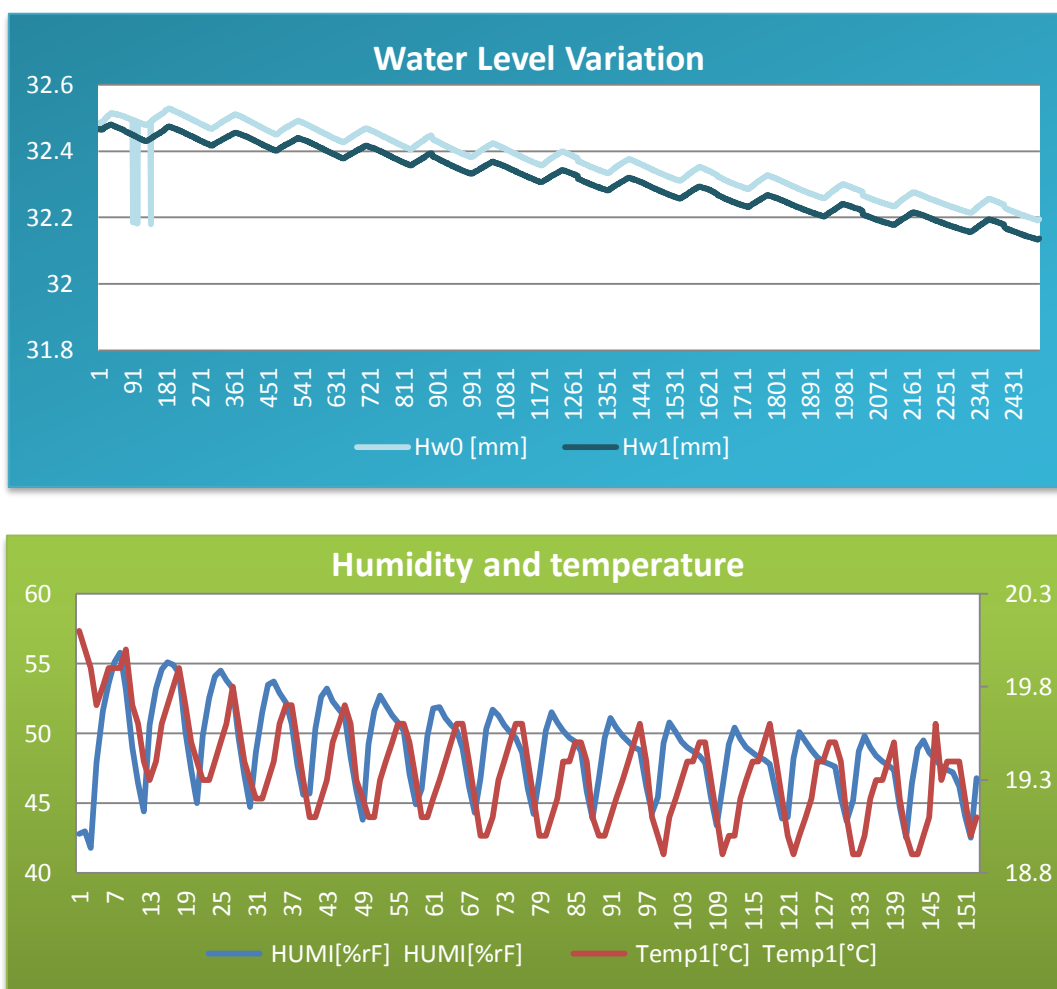


Figura 3.26 - Gráfico de Resultados de teste estático prolongado, apresentando variação do nível de líquido e evidenciando relação com ciclos de climatização.

O sistema foi colocado no Laboratório da *Survey Team*, sendo mantido fechado e com climatização regulada. As variáveis climatéricas foram então monitorizadas e registradas continuamente por um dispositivo *Thommen Meteo Station* ao longo de vários dias, interligado com o PXI.

Ao colocar o sistema num estado de testes contínuos permitindo monitorizar o nível de água em cada cápsula, podemos verificar claramente uma tendência e correlação entre os ciclos de climatização com as perdas no nível de água, responsáveis pelo efeito *ripple* visto.

Estes testes foram efectuados sob um período com a duração de 1 dia, ademais acrescidos de diversas repetições dos mesmos, e confirmando os resultados obtidos coerentemente.

Apesar de este fenómeno não ser ainda completamente explicável, é especulado a este ponto que a desgaseificação da água seja um dos actores deste efeito, já que uma perda de volume seria vista pelo sistema da mesma forma que uma perda de líquido.

3.10 – Tratamento de Erros – Central Error Handler

Embebida na arquitectura da aplicação está a estratégia de Tratamento de Erros passíveis de ocorrerem durante o funcionamento do sistema. Embora a aplicação deva ter o máximo de durabilidade, é sempre passível de ocorrerem erros que pelo menos devem ser registados. Como forma de “*manusear*” os mesmos, a infra-estrutura de captura dos Erros já foi descrita anteriormente até ao ponto destes serem encaminhados para o Módulo de Gestor de Erros, *Central Error Handler* - CEH.

Presente no *Error Case* de cada *Loop* principal, o *Error Handler – Minion* é encarregado de receber, formatar e comunicar o erro ao Módulo de Tratamento Central, e aguardar pela resposta bloqueando a execução. No caso de não a obter dentro de um tempo limite, ou caso nenhuma solução seja encontrada (erro desconhecido, por exemplo) a resposta padrão é a de registar o erro, a sua localização e descrição, e sair da aplicação o mais “graciosamente” possível. Este VI é corrido de forma *reentrante*, significando que cada chamada irá originar uma nova instância limpa de dados – um *clone*; caso contrário, o comportamento do VI seria o de manter apenas uma mesma instância activa a um dado momento, uma forma bloqueante de processamento. Deste modo, permitimos a resposta simultânea a erros que possam ocorrer ao mesmo tempo em diferentes pontos do sistema.

Cada erro possui um identificador único, constituído pelo seu Código de Erro aglomerado pelo nome do VI onde este ocorreu. Toda a informação do erro, e seu ID é encaminhada pelo *Minion* para o Módulo de tratamento central. Este irá verificar se existe algum registo correspondente no **ficheiro de descrição de erros** *HLS_Custom_Errors.txt* presente no directório da aplicação \Files\Custom Errors. Os conteúdos deste ficheiro, e a maneira como este é carregado são constituintes do processo de inicialização do módulo.



Figura 3.27 – Fluxograma de procedimento para o tratamento de Erros através do *Central Error Handler*

Caso o ID do Erro exista, o CEH irá então ler alguma informação extra que esteja registrada que possa ser útil apresentar ao utilizador. O seu próximo passo é verificar se existe um VI de resolução do problema identificado ou não, denominado de “**Fixer.VI**”. Os *Fixer* VIs são VIs que como a sua designação indica, possuem o código específico de tratamento para um dado erro, e localizam-se no directório de \Files\Custom Errors\Fixers. Este processo descrito encontra outra forma de expressão no fluxograma representado pela Figura 3.27

A vantagem de possuir estes VIs separados da integração rígida do projecto é que possibilita a expansão do processo de tratamento apelando à sua modularidade; se no futuro for desenvolvida a resolução de um erro que anteriormente era impossível ou desconhecida, basta colocar o *Fixer* contendo o código no directório correcto e o Módulo CEH irá procurar pelo mesmo e executar a operação de correcção correspondente.

3.11 – Theo Data Viewer

Nesta primeira fase de desenvolvimento, a mobilidade do sistema dependente do operador que posiciona o mesmo de ponto a ponto ao longo dos Ímanes Supercondutores do LHC. Cada Íman possui o mesmo tipo de Suportes de Encaixe ao longo de pontos pré-determinados, visualizado na Figura 3.28 permitindo a diversos equipamentos serem colocados. É precisamente nestes apoios, de dimensões conhecidas e registradas, que o sistema será inserido, montando as cápsulas com os encaixes apropriados.



Figura 3.28 – Suporte de Encaixe (esquerda) e sua localização típica num Íman dentro do LHC

Desta forma, ao estabelecer uma medição, é necessário identificar a referência do íman e a dos pontos a que se refere, constituindo o identificador único de cada ponto. Seguindo as normas do CCS, o formato dos dados resultantes da medição deve ser registrado segundo as normas correspondentes [11].

Tabela 3.4 – Formato dos dados A guardar segundo as normas do CCS

LIGNE DE NIVELLEMENT SFB SFB															
dans fichier de mesures terrain															
No ligne	Code ligne	Type mire	No mire	Lieu point	Type point	No point	Mire point	lecture (m)	Rallonge	Sens	Distance à la mire	Comment	Heure	Role	Opération terrain
I3	A2	A9	A6	A7	A6	A6	A1	F8.4	F8.5	A1	F8.3	A80	A8	A1	I2
(1,3)	(4,2)	(6,9)	(15,6)	(21,7)	(28,6)	(34,6)	(40,1)	(41,8)	(49,8)	(57,1)	(58,8)	(66,80)	(146,8)	(154,1)	(155,2)

A Tabela 3.4 representa não só os parâmetros a registar mas também, na sua primeira linha, o seu tipo e comprimento de caracteres máximo, com a segunda linha a indicar o seu índice e comprimento, respectivamente.

A leitura dos dados de posição é feita a partir de um ficheiro diferente – *Theo.dat*. Este ficheiro contém as características de cada ponto de mediação, como os valores de XYZ referidos ao CCS mas incluindo também o parâmetro H, que pretendemos medir com o sistema HLS.

A diferença entre os dois valores reside no facto de que este parâmetro foi medido inicialmente através de um **Nível Digital LEICA DNA03**, de precisão muito menor do que a atingida com o sistema HLS - entre 60 a 80 μm , dependendo do operador. Efectivamente, este valor ostenta uma diferença 10 vezes superior ao conseguido com o sistema HLS.

Não obstante, esta precisão é o suficiente para posicionar o sistema adequadamente dentro do seu alcance de operação de forma autónoma, como visto para o movimento ***Put in Place: Position from File***.

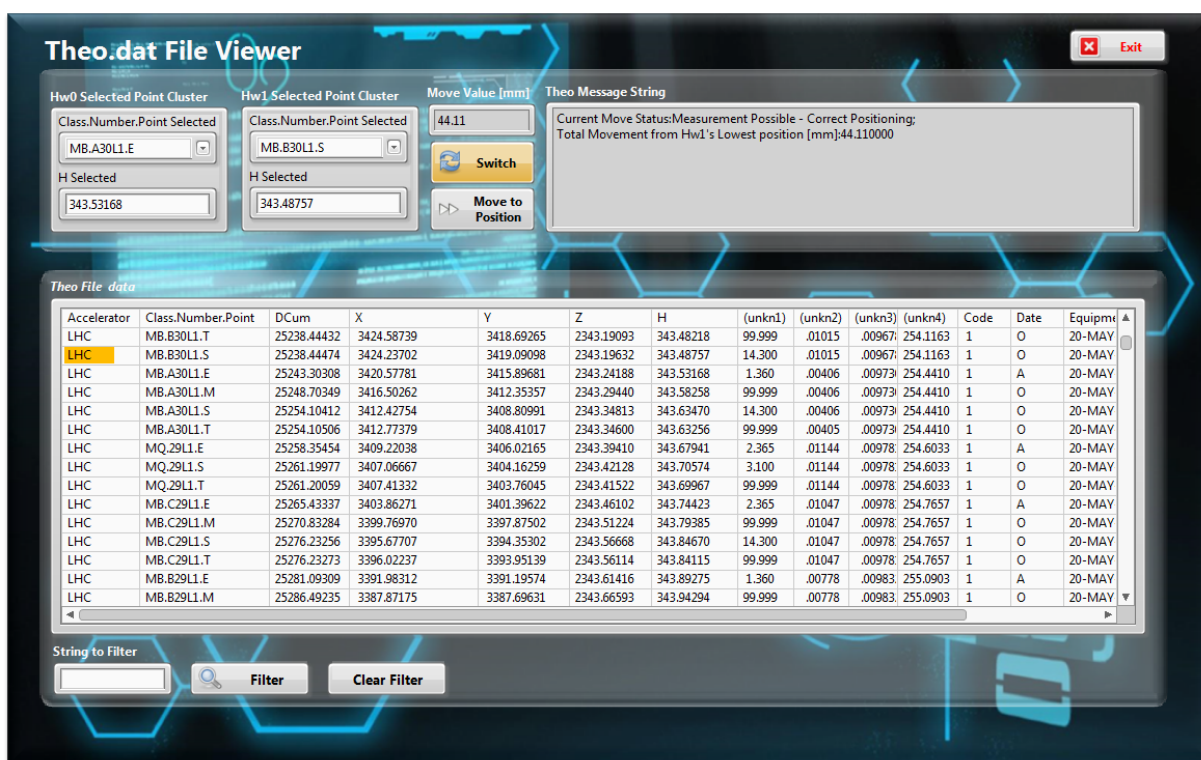
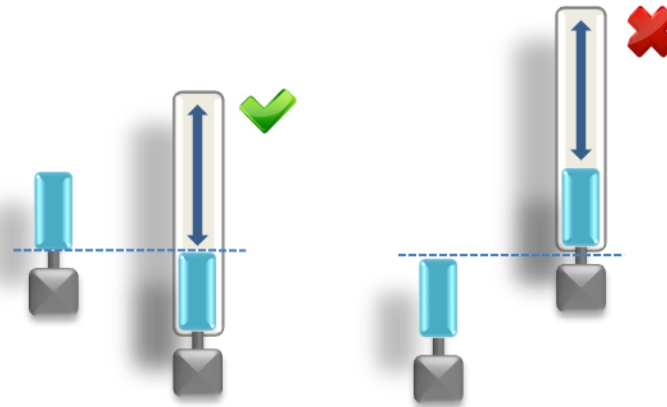


Figura 3.29 – GUI do Módulo *Theo Data Viewer*

O Módulo Visualizador dos Dados do Ficheiro Theo - *Theo Data Viewer*, cuja GUI é exposta na Figura 3.29, permite ao utilizador visualizar o ficheiro *Theo.dat* e seleccionar quais os pontos que serão medidos pelo sistema. Devido a apenas uma das cápsulas ser móvel (Hw1), se a cápsula estática for colocada num ponto que seja abaixo do ponto mais baixo passível de ser atingido pela cápsula móvel, será impossível colocar as duas cápsulas niveladas.

Exposto de outra forma, em concordância com a Figura 3.30, apenas se a cápsula móvel estiver colocada no ponto mais baixo será possível nivelar as duas. O Operador pode então iniciar o comando de *Move to Position*, que irá levar à sequência descrita anteriormente pelo processo *Put in Place: Position from File*.



Theo Data Viewer: Flowchart

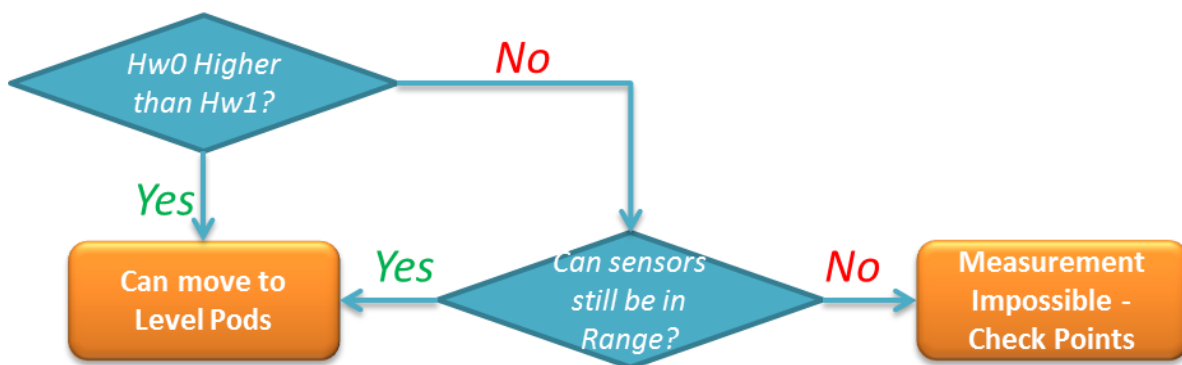


Figura 3.30 – Acima: Exemplo das possibilidades de movimentação perante o diferente posicionamento das cápsulas. Abaixo: Fluxograma de operação do *Theo Data Viewer*

Embora a lógica de controlo detecte se os pontos escolhidos pelo operador são passíveis de serem medidos ou não, não existe maneira de detectar se as cápsulas estão fisicamente montadas nos pontos correspondentes que o Operador declarou no programa. Neste ponto do projecto, em que a colocação das cápsulas nos pontos de medição é realizada manualmente pelo operador e que apenas uma das cápsulas é móvel, não existe alternativa além desta dependência da experiência do operador; contudo, no capítulo de desenvolvimentos futuros é abordada precisamente esta questão, bem como soluções passíveis de serem implementadas.

3.12 – Publicação via Web: Conexões Remotas

Como forma de potenciar a mobilidade do sistema integrado representado na Figura 3.31, desenhado a operar dentro de um espaço tão restrungido como os túneis do LHC, onde o espaço e a acessibilidade são escassos, foram desenvolvidas outras opções para o controlo remoto da aplicação.

Os módulos de *Hardware* do UHLS e do PXI foram integrados num só *rack*, mas continua a ser necessário um ecrã e periféricos que permitam a interface com o operador. Como forma de minimizar o equipamento transportado, duas principais soluções foram desenvolvidas: **publicação na web do Painel de Instrumentos, e uma aplicação para Tablet.**

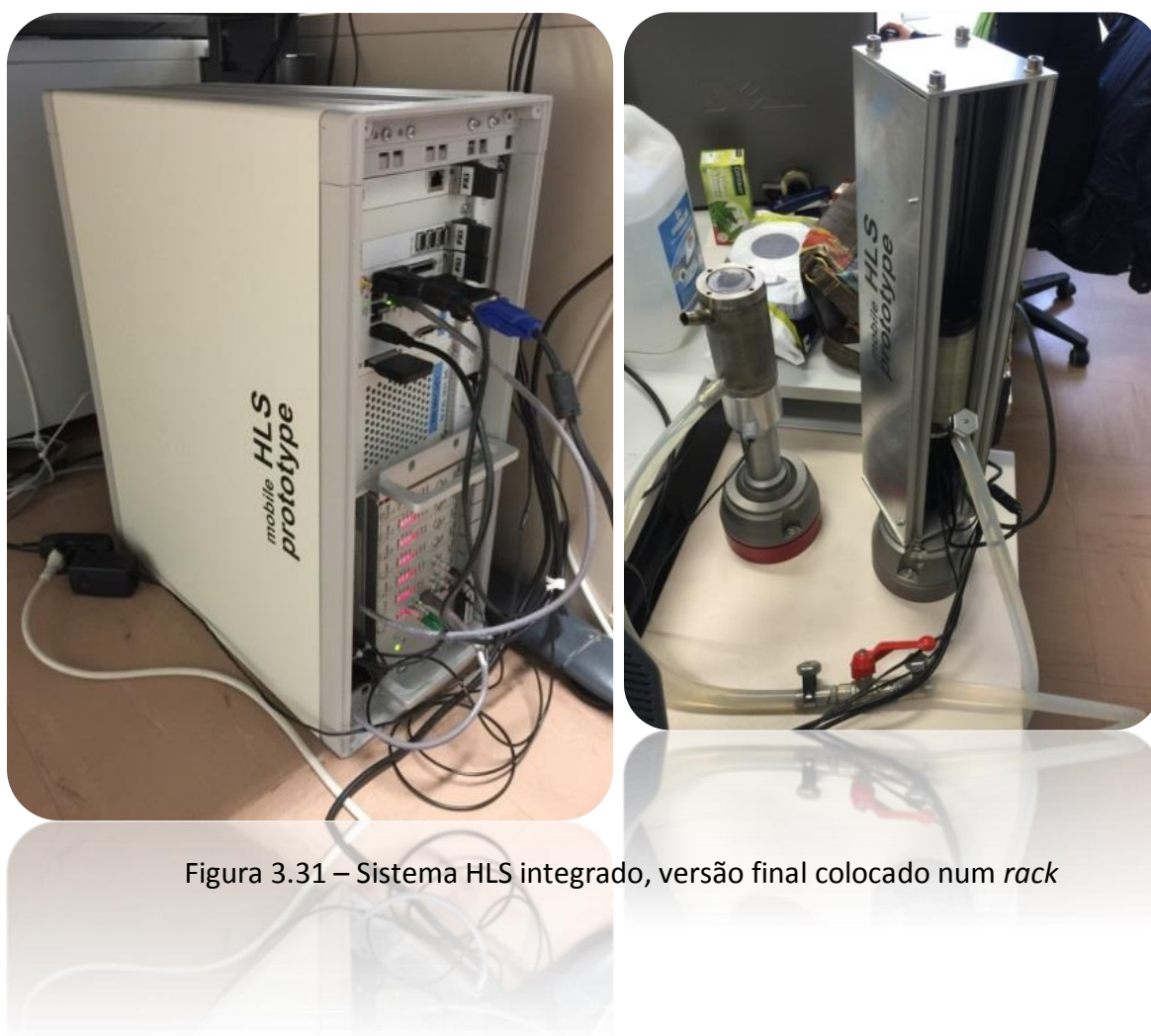


Figura 3.31 – Sistema HLS integrado, versão final colocado num rack

3.12.1 – Publicação Web do Front Panel

Através do seu *Web Server* embutido, o LabVIEW suporta nativamente uma opção de publicação online do Painel de Instrumentos dos seus VIs, possibilitando a um utilizador o controlo e leitura dos mesmos através de um browser remotamente, apenas necessitando de o *plug-in* LabVIEW *Real Time Engine*. Ao efectuar a configuração, à semelhança do exemplificado pela Figura 3.32, assumindo que os devidos processos, excepções de segurança e portas estão disponíveis da parte do servidor, apenas é necessário especificar quais os VIs a serem publicados e quais os endereços de IP (ou redes) que podem aceder ao mesmo. Posteriormente, o software gera o código *html* necessário, guardando o ficheiro *.xhtml* e possibilitando até a sua edição com ferramentas web mais avançadas. Para este projecto, foi determinado que apenas dispositivos dentro do domínio CERN possam aceder ao controlo remoto do HLS.

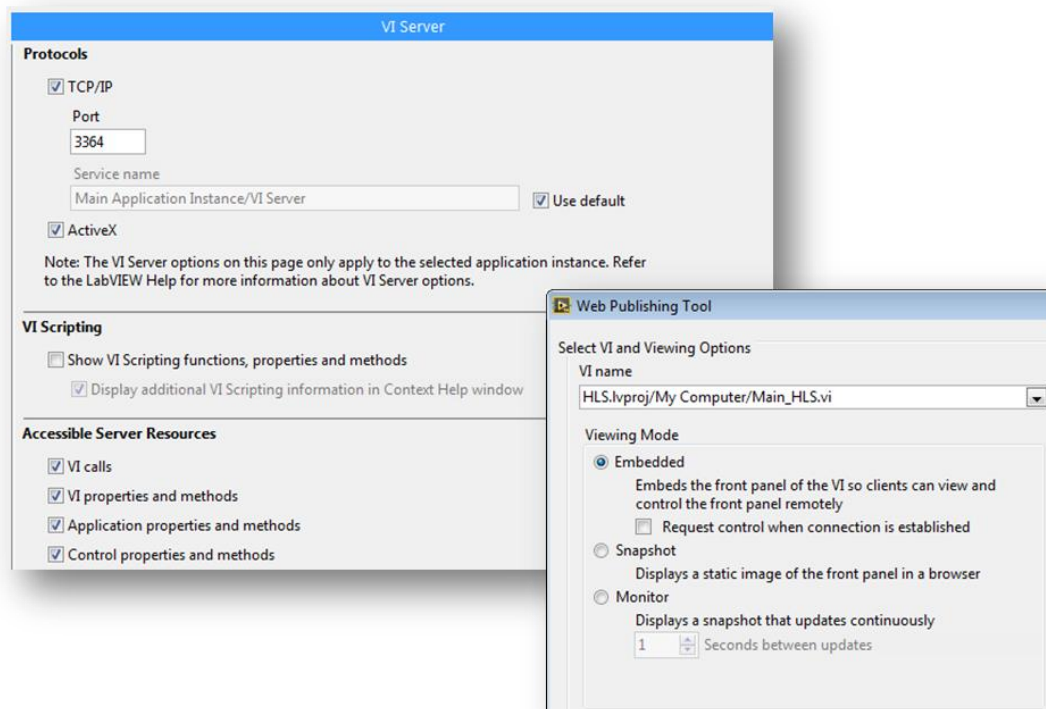


Figura 3.32 – Configurações exemplo para o *Web Server publisher*.

3.12.2 – Aplicação para Tablet

Devido ao facto de não existir o *plug-in* com o driver *LabVIEW RunTime Engine* para dispositivos que corram Android ou iOS, a utilização do sistema através de uma conexão com uma Tablet torna-se mais complicada e não prática, sendo que a solução passaria por uma *tablet* Windows com LabVIEW e IE, ou através de uma conexão remota com o dispositivo. Para abordar esta questão, foi utilizado o *software* da NI Data *Dashboard*, que permite o desenvolvimento de Painéis de Instrumentos bastante básicos que possam intercomunicar com o servidor através de variáveis partilhadas – *Shared Variables*, ou subscrevendo uma publicação *web-service* da aplicação.

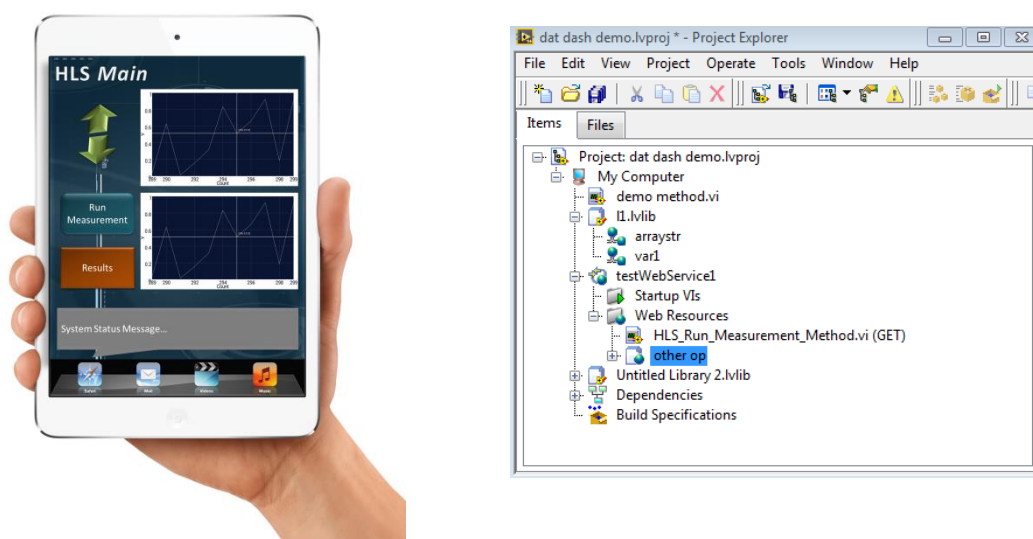


Figura 3.33 – GUI da aplicação *Data Dashboard* (esquerda) e representação dos métodos na janela de projecto LabVIEW (direita)

A primeira abordagem através de *Shared Variables* implica um *polling* constante a uma dada periodicidade, sondando o estado de uma variável correndo no código. Ao subscrever um serviço, estaremos apenas a receber *updates* quando o código respectivo activasse o método em questão. O princípio de funcionamento é então o envio e recepção de informação através de métodos que são previamente declarados na janela de projecto em LabVIEW com um determinado grupo de parâmetros.

No momento de elaboração deste Relatório, apenas os métodos que permitem executar um movimento de 1mm para cima ou para baixo da cápsula móvel, bem como executar uma medição estão disponíveis e testados, conforme visualizado na Figura 3.33, com os planos para desenvolvimento das outras funcionalidades sendo adiados pelo cronograma do projecto até Janeiro de 2015.

3.13 – Módulo de Testes

Como forma de suportar os testes operacionais de maneira mais eficiente, foi desenvolvido um módulo de testes capaz de emular o comportamento de certas partes do sistema e de exibir comportamentos de medição diferentes.

O Módulo de Testes - *Tester*, com a sua GUI e princípio de funcionamento expostas pela Figura 3.34, contém uma decomposição de todas as ações elementares que o sistema pode interpretar, estabelecidas dentro de um sub-VI no formato de máquina de estados denominado *Tester Kinematics*. À semelhança de outros módulos, apenas necessitamos de fornecer ao mesmo a sequência de ações indicada para realizar um determinado teste. Graças a esta plasticidade, é possível combinar os elementos por forma a desenvolver qualquer tipo de teste, se necessário futuramente.

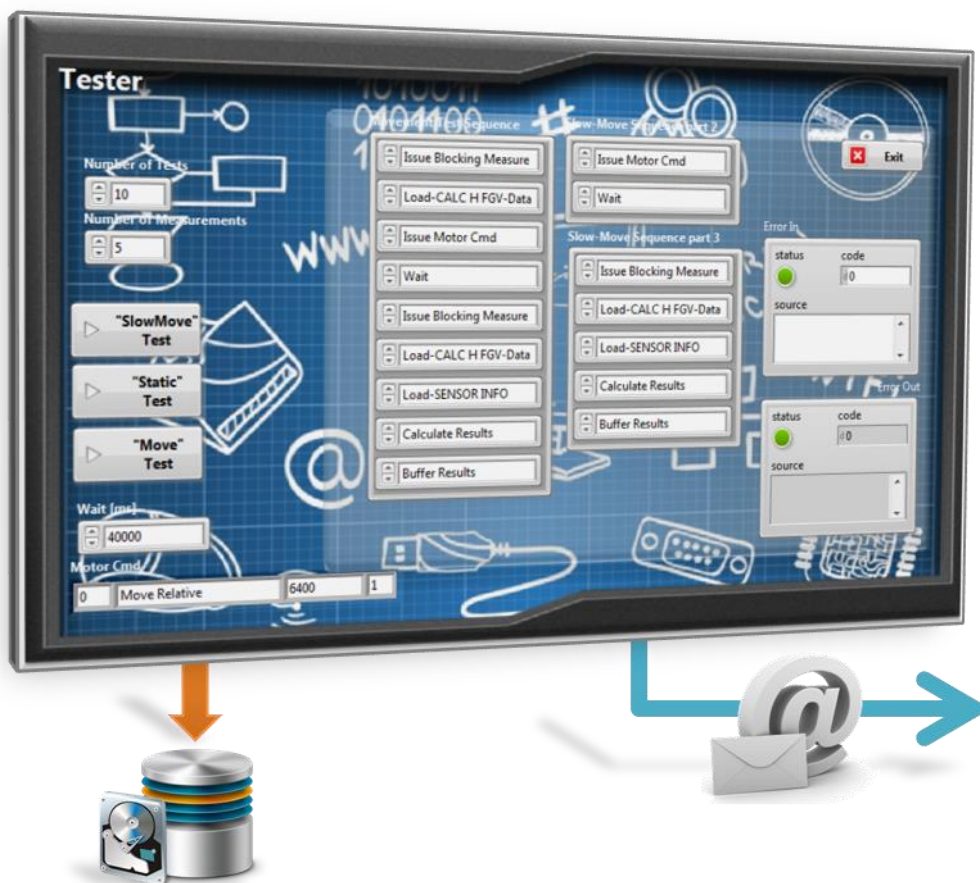


Figura 3.34 – GUI do Módulo *Tester* e representação dos seus canais de comunicação

Correntemente, três tipos de teste foram estabelecidos, cuja sequência e parâmetros são obtidos através do Painel de Instrumentos do Módulo, com os valores de defeito a serem carregados a partir do *Macros Manager*:

- **Teste Estático:** Estabelece medições contínuas do nível da água em ambas as cápsulas
- **Teste Movimento:** Segue uma sequência de *Medição->Movimento->Medição*. Indicado para testar o alcance real operacional dos sensores por nível de água contido nas cápsulas.
- **Teste Movimento-Lento:** seguindo a mesma sequência, mas realizando uma série de medições entre movimentos. Indicado para determinar o tempo de estabilização necessário e precisão entre movimentos.

Ao contrário dos Logs de operação, os resultados de testes são guardados em ficheiros Excel dentro da localização `\Files\Logs\Test Results`. Estes ficheiros são também enviados automaticamente em anexo por correio electrónico, cujo assunto é dado pela concatenação do nome do teste realizado e o seu *timestamp*.

CAPÍTULO 4 – ANÁLISE DE RESULTADOS

4.1 – Resultados Laboratoriais

Intercalando as duas etapas finais de desenvolvimento da aplicação sucedeu-se os resultados operacionais finais num ambiente controlado, dentro do Laboratório de Metrologia, Ed.926. Aqui o sistema foi colocado numa mesa de plano de mármore própria nivelada, retratada na Figura 4.1, e sob situações climatéricas controladas.

Os principais objectivos destas primeiras sessões de testes eram determinar, entre outros, os seguintes parâmetros:

- **Precisão** do Sistema;
- **Parâmetros de Configuração** (offsets, tolerâncias);
- **Alcance de Funcionamento** (em relação ao nível de água);
- **Stress Testing**: Investigação de comportamentos anormais e *debugging*;



Figura 4.1 – Sistema HLS durante os testes de operação fina no Laboratório de Metrologia

Outros tipos de testes realizados, pertinentes a aspectos mais práticos de operação, incluem a determinação do *offset* absoluto entre os dois suportes das cápsulas e a imunidade do sensor à inclinação do líquido dentro da cápsula. Estes últimos foram realizados com auxílio de um nível de inclinação de plano de precisão, conforme observado na Figura 4.2.



Figura 4.2 – Pormenor do sistema HLS durante testes de funcionamento em inclinação (esquerda) e estação meteorológica *Thommen* para registo das variáveis ambientais

Através da utilização das funcionalidades do Módulo *Tester*, além do fenómeno da perda do volume de líquido total foi descoberta a ocorrência de um outro fenómeno recorrente, onde a leitura de resultados do Sensor da cápsula 0 demonstra, periodicamente, resultados inválidos, ademais possuindo com um valor de módulo relativamente constante. Estas ocorrências, de real causa indeterminada até ao momento, encontram-se identificadas pelos círculos vermelhos no gráfico apresentado na Figura 4.3, representando medições contínuas estáticas sob um período prolongado.

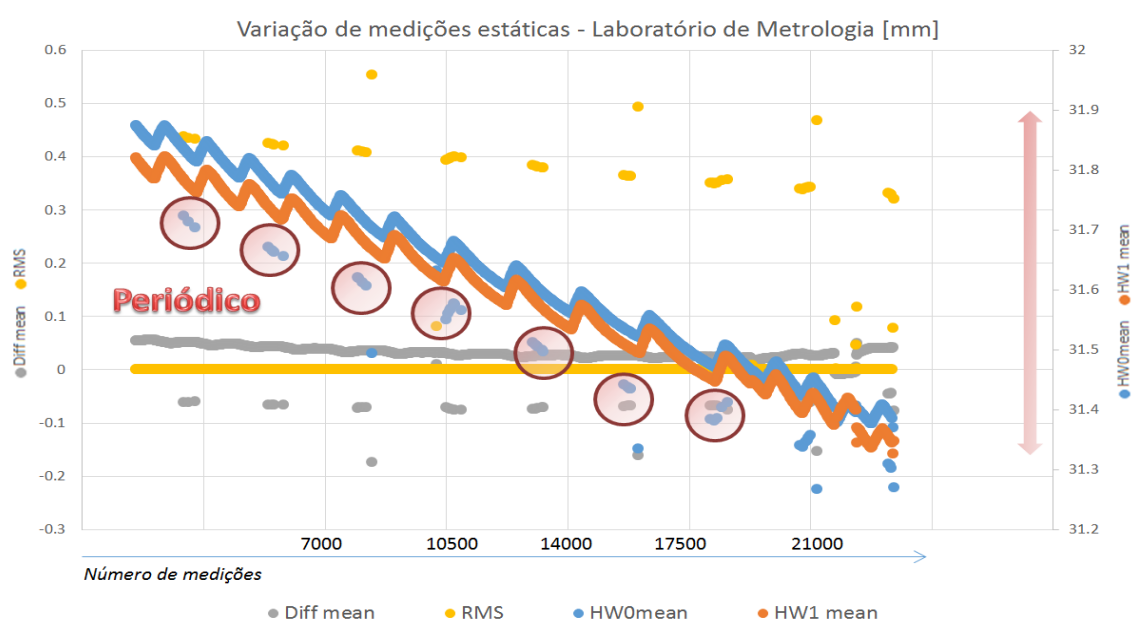


Figura 4.3 – Comportamento HLS durante medições prolongadas.

Os gráficos apresentados na Figura 4.4 representam os conjuntos de medições a partir de uma sequência de operação contínua. No gráfico superior, constituindo uma única medição, elaborada por uma amostra de 20 pontos, é possível observar que a variação total máxima entre os pontos não excede os 2 μ m, com uma média de 0,29 μ m. Para o gráfico inferior sendo que cada medição tem uma duração aproximada de 40s, o fenómeno de perda de líquido ganha relevância e é facilmente identificado; contudo, a precisão do sistema não excede uma variação de 10 μ m para uma amostra de 100 medições.

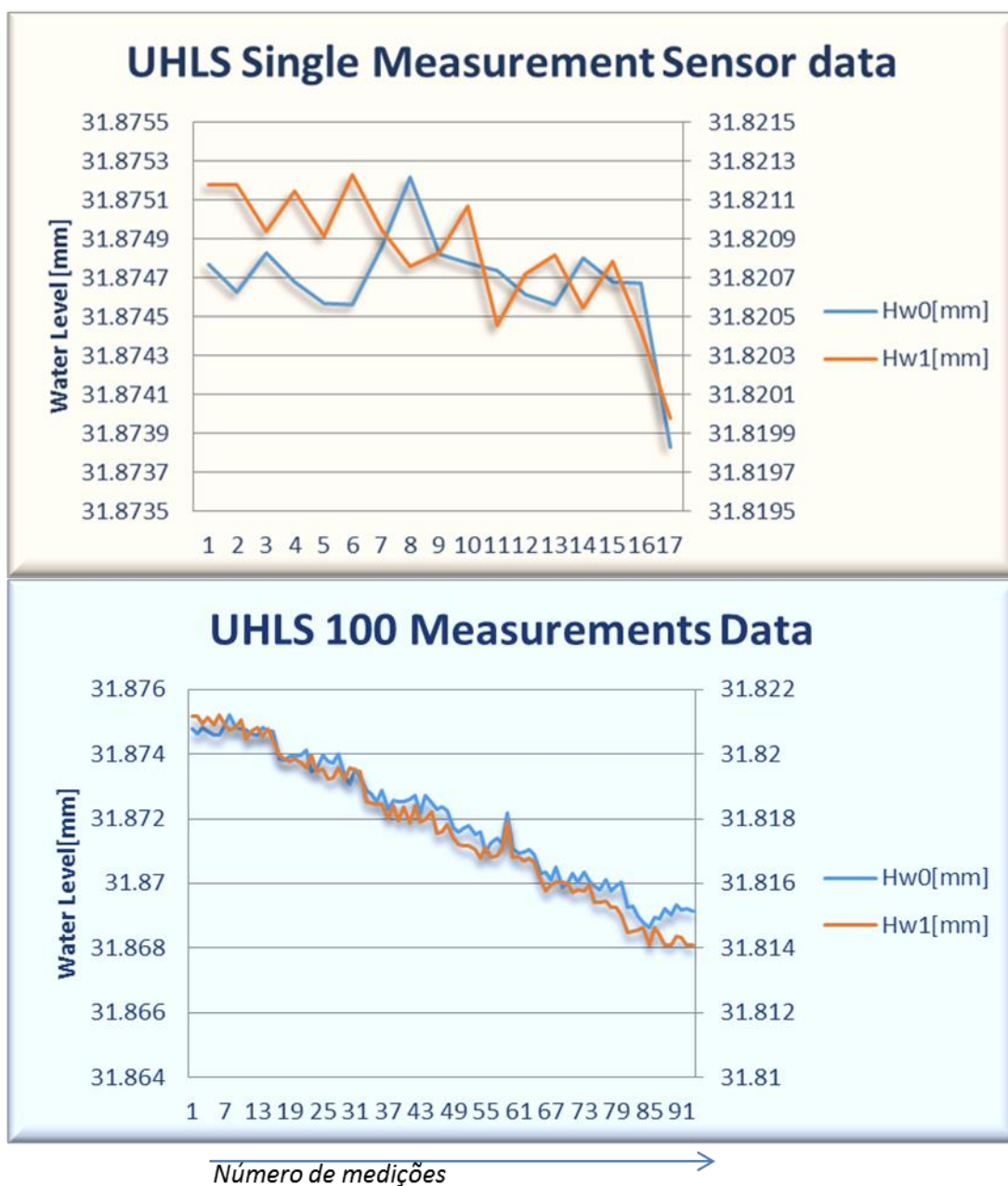


Figura 4.4 – Gráficos de Resultados de precisão: Série única (acima) e sequência de 100 séries (abaixo).

Os parâmetros de calibração que foram determinados com recurso a este módulo podem então ser sumarizados na Tabela 4.1. A partir dos testes de medição é possível determinar a **Precisão Média**. O **Offset Absoluto** refere-se à diferença de altura entre as cápsulas quando colocadas na posição 'zero' (inicial), devido às diferenças nas suas extensões e suportes. A **Inclinação Máxima** é definida através dos testes de funcionamento com a utilização do plano de inclinação de precisão, e o **Alcance Operacional** através dos procedimentos de *Teste – Movimento*, implementados pelo Módulo de Testes. Finalmente, o **Tempo de Estabilização do Líquido** e o **Coefficiente de Correlação** referente à variação do líquido entre as duas cápsulas foram determinados empiricamente através da análise de uma considerável amostra de testes.

Tabela 4.1 – Parâmetros de operação empiricamente determinados

Parâmetro:	Valor:
Precisão Média:	$\pm 2\mu\text{m}$
Offset Absoluto entre Cápsulas:	140,986[mm]
Inclinação Máxima:	20 [mRAD]
Alcance Operacional:	[18;45]mm
Tempo de Estabilização do líquido:	40s
Coefficiente de Correlação:	0,6

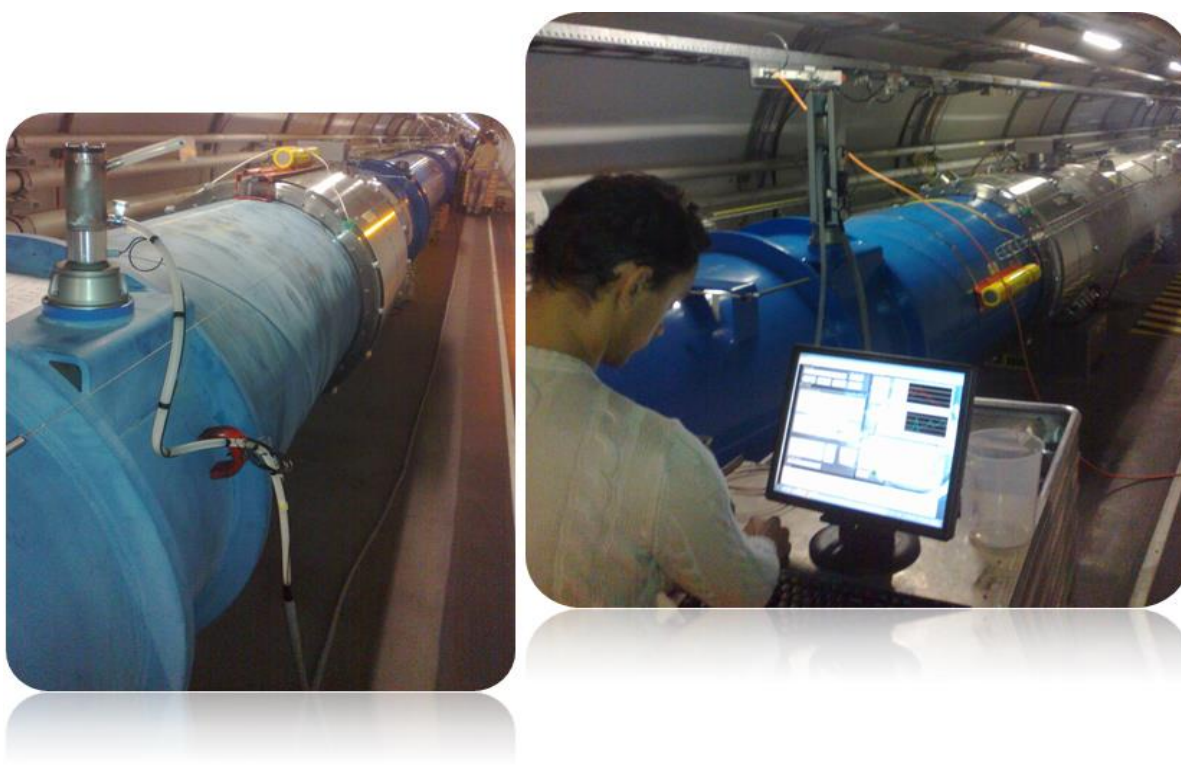
Sendo que a altura total do ponto de medição, convencionalmente o ponto de localização da cápsula imóvel H_{w0} , é obtida através da expressão (4.1):

$$Total\ H = Optical\ Scale + |\Delta H_w| \quad (4.1)$$

Desta forma, observando as medições efectuadas, podemos corroborar a precisão de todo o sistema dentro de uma base empírica que excede largamente os requisitos e expectativas do sistema quanto à operação de medição da diferença de altura entre os pontos.

4.2 – Resultados Operacionais: LHC

Tendo comprovado a precisão do sistema, satisfazendo a primeira parte das especificações técnicas, restava apenas testar a mobilidade e praticidade de execução do sistema no seu ambiente de operação, o LHC. Para tal, juntamente com os Supervisores Cédric Charrondiérre e Patrick Bestmann foi permitido o acesso ao LHC - Ponto 5, no início de Dezembro de 2014. Dentro destes factores, era necessário discernir os tempos de operação e manuseamento do sistema, incluindo o seu transporte, tempo de colocação das cápsulas, tempos de estabilização e medição e finalmente o tempo de resolução de problemas no local, em caso de alguma situação irregular como a existência de bolhas de ar no tubo conector entre as cápsulas, impedindo a circulação do líquido.



Foram efectuadas medições numa sequência de 12 ímanes, à semelhança do observado na Figura 4.5, onde é possível observar o sistema HLS já posicionado e em operação numa extensão máxima do seu alcance entre pontos de 15m. A Figura 4.6 demostra o transporte do sistema HLS móvel dentro de um contentor próprio equipado com uma fonte de alimentação ininterrupta (por forma a potenciar uma maior flexibilidade e capacidade de operação), juntamente com o processo de um outro tipo de medição, sobre o eixo de um íman Quadripolo.



O tempo total de medição de um ponto é estimado em perto de 6 minutos, valor considerado bastante positivo, levando em conta um factor de tolerância de 10% por cada ponto a ser medido, devido a circunstâncias inesperadas; de facto apenas foi necessário drenar as bolhas de ar presentes no tubo aquando da sua primeira execução, algo perfeitamente dentro do esperado e resultante do enchimento do próprio tubo com água.

Note-se que embora o teste prático em ambiente real, com a avaliação dos valores obtidos e maneabilidade geral do sistema a ser bastante positiva, é necessário levar em consideração que a equipa utilizada possui 3 membros, um a mais do que normalmente ocorre em grandes campanhas de medições no LHC e por operadores experientes com conhecimentos aprofundados do sistema.

CAPÍTULO 5 – CONCLUSÕES

5.1 – Conclusões

O desenvolvimento deste projecto de um sistema de Nivelamento Hidrostático Móvel ostentava intrinsecamente uma multitude de especificidades ditada pela natureza do ambiente da sua aplicação final – o LHC. Tanto os requisitos de precisão como de portabilidade e exequibilidade da operação deviam ser respeitados conjugadamente, com a adição de serem enquadrados na janela temporal de execução do *Long Shutdown 1*, impossível de ser alterada.

A necessidade da medição do nível dos Ímanes Supercondutores é uma medida imperativa no funcionamento e operação dos Aceleradores, um ambiente com necessidades bastante exigentes e muito próprias, tornando-se natural que os sistemas utilizados sejam muitas vezes representados por soluções customizadas.

Conforme referido, o sistema possui a sua constituição baseada em 4 elementos de *hardware* preponderantes: a mesa de translação, a régua óptica, o sensor *UHLS* e o *PXI*.

A implementação da aplicação foi largamente acelerada pela utilização do *software* LabVIEW, definitivamente a ferramenta apropriada para a unificação e controlo de diferentes dispositivos e componentes de *hardware*, cada um com os seus protocolos de comunicação distintos. Através desta plataforma, foi possível estabelecer não só uma lógica de controlo de alto nível mas também o processamento e tratamento dos dados, facilitando também a interligação de várias bibliotecas com diferentes formatos.

Podemos decompor o projecto em duas fases distintas: a primeira etapa, contemplando o desenvolvimento dos *drivers* para o *hardware* separados, formando uma *proof of concept* da funcionalidade dos mesmos, e uma segunda parte que consistiu na integração dos mesmos e desenvolvimento da lógica de controlo e estrutura da aplicação.

Embora atingir os níveis de precisão desejados não constituiu nunca um problema com a tecnologia e sistema utilizados, a coerência e viabilidade das medições revelou-se difícil de discernir. Vários factores apresentaram-se como determinantes para o bom funcionamento do sistema, como o alinhamento da peça de referência do sensor ultra-sónico, o tempo de estabilização do líquido ou mesmo aspectos construtivos como a assimetria das cápsulas. Como resposta, o tratamento estatístico e de diagnóstico inserido na aplicação automatizam ao máximo a validação da medição, prestando um nível de imunidade e autocorreção a estes factores, libertando o Operador de possuir experiência especializada e executar decisões sobre a mesma, pavimentado o caminho para os desenvolvimentos futuros que colocam o sistema num nível completamente autónomo.

Com medições da ordem de precisão bem acima do esperado situando-se na ordem de grandezas de poucos micrómetros, um tratamento de dados autónomo e eficiente que

permite ultrapassar as questões e debilidades da escolha tecnológica, de mobilidade comprovada e inserido numa arquitectura de aplicação sólida, eficiente e preparada para desenvolvimentos futuros, o projecto é amplamente considerado um sucesso exemplar por ambas as secções e grupos intervenientes.

É, contudo, o resultado final que porventura constitui o maior indicador do nível de sucesso do projecto, que, desenvolvido segundo os objectivos planeados, foi implementado atempadamente e permitirá realizar uma campanha de medições a tempo do término do LS1 e servirá de apoio à operação do LHC durante a sua próxima campanha de operação.

5.2 – Desenvolvimentos Futuros

Com o sucesso da campanha de medições sobre o LHC, o projecto HLS será continuado numa vertente de expandir a sua mobilidade e facilidade de operação. Conforme referido, é esperado que o sistema integre o *Survey Train*, sendo montado no mesmo quando desejado, e que efectue as medições de forma autónoma. Para tal está prevista a utilização de um braço robótico que colocara as cápsulas nos pontos adequados, equipado com uma câmara por forma a reconhecer os mesmos e efectuar a cinemática de movimentos adequada, à semelhança da representação do sistema observada na Figura 7.1. Sendo que as duas cápsulas serão tornadas móveis, o braço irá colocar ambas nos pontos destinados, efectuar a medição e recolher as mesmas prosseguindo para o próximo ponto de medição.

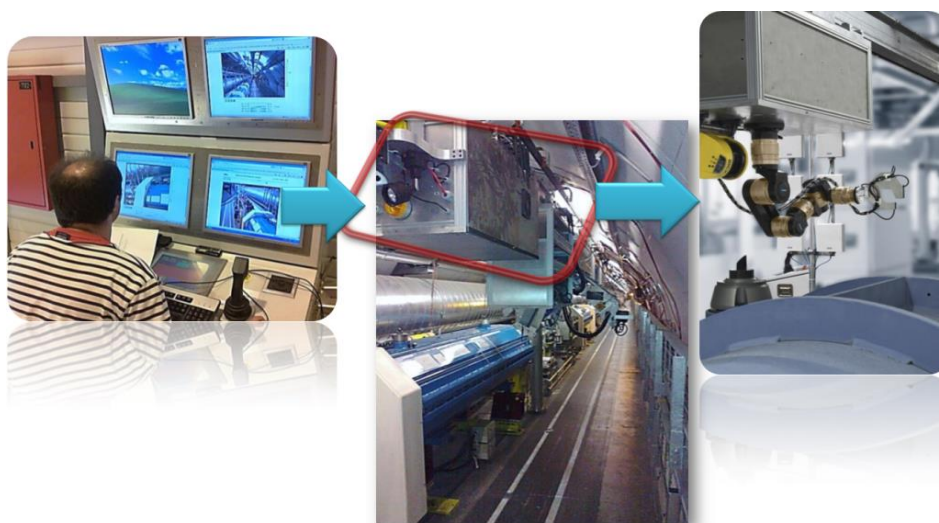


Figura 7.1 – Representação Futura do sistema HLS de Medições

Este projecto, contemplando uma colaboração especial entre 3 Grupos, juntando-se uma secção de especialistas de robótica, encontra-se muito recentemente aprovado para implementação ao longo de um plano de 3 anos, previsto para entrar em funcionamento durante o próximo período de paragem, o *Long shutdown 2* em 2018.

Referências Bibliográficas

- [1] – CERN Communications Group, “*Rapport Annuel-2013*”, 2013, versão electrónica acedida em: 10/12/2014 em: <http://publications.cern.ch/>
- [2] – CERN Communications Group, “*LHC: The Guide*”, 2009, versão electrónica acedida em: 10/12/2014 em: <http://cds.cern.ch/collection/Multimedia%20%26%20Outreach?ln=en>
- [3] – ALTARELLI, G., “*Theoretical implications of the LHC Results*”, Nuclear Instruments and Methods in Physics Research A 742 (2014)56-62, 2014
- [4] – OLIVE, K.A. et al., “*Review of Particle Physics*”, Chinese Physics C N°9(2014) 090001, 2014
- [5] – MISSIAEN, D., “*Geodesy and Alignment for Particle Accelerators*”, Accelerator and Detectors, Springer Berlin Heidelberg, 10.1007/978-3-642-23053-0_33, 2013
- [6] – JONES, M., “*Des Éléments Importants des Systèmes de Référence et de la Géodésie au CERN*”, 2014, disponível em versão electrónica acedida a: 10/12/2014 em <http://indico.cern.ch/event/306787/material/slides/1.pdf>
- [7] – BESTMANN, P., et al, “*Ultrasonic Hydrostatic Levelling System project Guidelines*”, CERN, 2013
- [8] – PRENTING, J., SCHLÖSSER M., “*High Precision Survey and Alignment of Large Linear Accelerators*”, 22nd International Linear Accelerator Conference Lubeck, Alemanha, Agosto 2004
- [9] – CHARRONDIÈRE, C. et al, “*Remote Control of Heterogeneous Sensors for the 3D LHC Collimator alignment*”, 14th International Conference on Accelerator & Large Experimental Physics Control Systems, San Francisco, CA, Outubro 2013
- [10] – SCHLÖSSER M., REINECKE M., “*Hydrostatic Levelling System Metering Point Electronics 2 Operating Manual, Version 1.2*”, DESY, Dezembro 2013
- [11] – PODEVIN C., “*SBB Format des Fichier de SFB/SFBI*”, CERN, 2007